

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE GRADUAÇÃO DE ENGENHARIA CIVIL**

**MODELO DE KRIGAGEM
PARA CONFIABILIDADE**

GUSTAVO DELA BRUNA NORONHA

Trabalho de Conclusão de Curso

Florianópolis/SC

2016

GUSTAVO DELA BRUNA NORONHA

**MODELO DE KRIGAGEM
PARA CONFIABILIDADE**

Trabalho de Conclusão de Curso apresentado
à Banca Examinadora do Curso de Graduação
de Engenharia Civil da Universidade Federal
de Santa Catarina como requisito parcial para
obtenção do Grau de Engenheiro Civil.
Professor Orientador: Rafael Holdorf Lopez.

Florianópolis/SC

2016

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Noronha, Gustavo Dela Bruna

Modelo de Krigagem para confiabilidade / Gustavo Dela
Bruna Noronha ; orientador, Rafael Holdorf Lopez -
Florianópolis, SC, 2016.

69 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico.
Graduação em Engenharia Civil.

Inclui referências

1. Engenharia Civil. 2. Estruturas. 3. Confiabilidade.
4. Simulação de Monte Carlo. 5. Krigagem. I. Lopez, Rafael
Holdorf. II. Universidade Federal de Santa Catarina.
Graduação em Engenharia Civil. III. Título.

GUSTAVO DELA BRUNA NORONHA

**MODELO DE KRIGAGEM
PARA CONFIABILIDADE**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso de Graduação de Engenharia Civil da Universidade Federal de Santa Catarina como requisito parcial para obtenção do Grau de Engenheiro Civil.

TRABALHO APROVADO PELA COMISSÃO EXAMINADORA
EM FLORIANÓPOLIS, 29 DE JUNHO DE 2016

Prof. Luis Alberto Gómez, Dr.
Coordenador do Curso

Profa. Lia Caetano Bastos, Dr.
Coordenador de TCC



Prof. Rafael Holdorf Lopez, Dr.
Orientador

Prof. Wellison José de Santana Gomes, Dr.
Examinador

Felipe Carraro, Me.
Examinador

AGRADECIMENTOS

Agradeço inicialmente aos meus pais Alan Noronha e Magali Dela Bruna Noronha pelo apoio e carinho que tem tido comigo, fazendo esforços sem medidas e incentivando sempre que possível. Aos meus irmãos Henrique e Thiago por poder contar com vocês sempre, independente da situação. Ao meu vô Luiz pelo exemplo de dedicação e trabalho duro, a minhas avós Celina e Glória pelo carinho sem igual. À minha tia Deise pela dedicação e cuidado, e principalmente para minha tia e madrinha Denise por todo o carinho, inclusive por me ensinar matemática quando mais novo. E agradeço também a toda minha família, seria impossível citar todos aqui, mas obrigado pelo carinho.

Gostaria também de agradecer os meus professores, que ajudaram na construção dos conhecimentos que tenho hoje. Em particular ao professor Rafael Holdorf Lopez por me orientar neste trabalho e ajudar a compreender os assuntos envolvidos, e principalmente pois com as atividades propostas por ele incentivarem de maneira a me manter firme no curso de engenharia civil.

Aos membros do CORE, em especial os professores Leandro Miguel, Rafael Holdorf e Wellison Gomes, assim como aos integrantes Rafael Rodrigues Souza e André Gustavo Carlon. O aprendizado que foi possível por meio deles foi de grande importância tanto para este trabalho, quanto para minha compreensão da engenharia.

Também aos meus amigos pela paciência e compreensão nesses últimos meses do qual fiquei muito ausente, mas vocês sempre estavam lá para me dar apoio quando precisei.

Por fim, mas não menos importante à minha namorada Ana Paula. Sem o seu esforço muitos dos trabalhos que realizei não ficariam tão bons, você sempre estava lá para ouvir meus desabafos, dar atenção e muito carinho. Obrigado por tudo.

“A faca do caçador não esculpe o próprio cabo.”
(Ditado coreano)

RESUMO

Considerando a importância da simulação de Monte Carlo para estudos de confiabilidade e com o objetivo de reduzir o custo computacional, este trabalho foca-se no estudo do uso de um meta modelo para Monte Carlo. A meta modelagem serve para estimar os resultados de um sistema por meio de equações, correlacionando pontos conhecidos, chamados de pontos de controle, com os pontos desconhecidos, sendo neste trabalho os pontos amostrais de Monte Carlo. Este trabalho utilizou-se dos estudos de Echard, Gayton e Lemaire sobre krigagem para confiabilidade com o objetivo de estudar, melhorar e explicar o uso da krigagem para confiabilidade de forma compreensível. Os resultados indicaram que as alterações foram úteis, o programa desenvolvido funcional e o uso de meta modelos para cálculo da confiabilidade por meio da simulação de Monte Carlo promissor.

Palavras-chave: Krigagem; Kriging; confiabilidade; Meta modelagem. simulação de Monte Carlo.

ABSTRACT

Considering the importance of Monte Carlo simulation for reliability analysis and in order to reduce its computational cost, this study focuses on the use of a metamodel with Monte Carlo simulation. Metamodels estimate the results of a system by equations correlating known points, called control points, with the unknowns, which in this paper are the Monte Carlo sampling points. This study used the study of Echard, Gayton and Lemaire on kriging for reliability in order to study, improve and explain the use of kriging for reliability on a comprehensible way. The results of this paper indicated that the improvements were helpful, the program developed was functional and that the use of metamodels for calculation of reliability by Monte Carlo simulation is promising.

Key-words: Kriging; Reliability; Metamodeling; Monte Carlo Simulation.

LISTA DE ILUSTRAÇÕES

Figura 1 – Distribuição uniforme	22
Figura 2 – Distribuição normal	22
Figura 3 – Distribuição lognormal	23
Figura 4 – Distribuição de Gumbel	23
Figura 5 – Exemplo 1 de limite de falha	25
Figura 6 – Exemplo 2 de limite de falha	26
Figura 7 – Comparação de tempo de execução para exemplo 3	38
Figura 8 – Esquema do programa desenvolvido	40
Figura 9 – Visualização do exemplo 3.	43
Figura 10 – Teoria clássica de laminados.	44
Figura 11 – Visualização do exemplo 4.	45
Figura 12 – Resultado do modelo para o caso 1. Parte 1	48
Figura 13 – Resultado do modelo para o caso 1. Parte 2	49
Figura 14 – Probabilidade de falha estimada para o caso 1.	49
Figura 15 – Resultado do modelo para o caso 1. Parte 3	50
Figura 16 – Probabilidade de falha estimada para o caso 2.	51
Figura 17 – Resultado do modelo para o caso 2. Parte 1	52
Figura 18 – Resultado do modelo para o caso 2. Parte 2	53
Figura 19 – Probabilidade de falha estimada para o caso 3.	54
Figura 20 – Resultado do modelo para o caso 3. Parte 1	55
Figura 21 – Resultado do modelo para o caso 3. Parte 2	56

LISTA DE TABELAS

Tabela 1	– Valores para o método de Monte Carlo do exemplo 1.	42
Tabela 2	– Valores para o método de Monte Carlo do exemplo 2.	43
Tabela 3	– Valores para variáveis aleatórias do exemplo 3.	45
Tabela 4	– Valores para o método de Monte Carlo do exemplo 3.	45
Tabela 5	– Valores para variáveis aleatórias do exemplo 4.	46
Tabela 6	– Valores para o método de Monte Carlo do exemplo 4.	46
Tabela 7	– Dados do programa com krigagem. Exemplo 1, semente 0, Ulim 1,5. .	57
Tabela 8	– Dados do programa com krigagem. Exemplo 1, semente 3, Ulim 1,5. .	58
Tabela 9	– Dados do programa com krigagem. Exemplo 1, Variando Ulim	59
Tabela 10	– Dados do programa com krigagem. Exemplo 2.	60
Tabela 11	– Dados do programa com krigagem. Exemplo 3, <i>Seed</i> 0.	61
Tabela 12	– Comparação para o exemplo 3.	61
Tabela 13	– Comparação para o exemplo 4.	64

SUMÁRIO

1 INTRODUÇÃO	15
1.1 JUSTIFICATIVA	15
1.2 OBJETIVOS	16
1.2.1 OBJETIVOS ESPECÍFICOS	16
1.3 ESTRUTURA DO TRABALHO	16
2 CONCEITOS BÁSICOS	17
2.1 ESTATÍSTICA E PROBABILIDADE	17
2.2 TEORIA DA PROBABILIDADE	18
2.2.1 DEFINIÇÃO CLÁSSICA	18
2.2.2 DEFINIÇÃO FREQUENTISTA	18
2.2.3 DEFINIÇÃO BAYESIANA	19
2.2.4 DEFINIÇÃO AXIOMÁTICA	19
2.2.5 FUNÇÕES DE PROBABILIDADE E DENSIDADE DE PROBABILIDADE	19
2.2.6 CONCEITOS DAS VARIÁVEIS ALEATÓRIAS	20
2.3 DISTRIBUIÇÕES RELEVANTES	21
2.3.1 DISTRIBUIÇÃO UNIFORME	21
2.3.2 DISTRIBUIÇÃO NORMAL	22
2.3.3 DISTRIBUIÇÃO LOGNORMAL	23
2.3.4 DISTRIBUIÇÃO DE GUMBEL	23
2.4 PROCESSOS ESTOCÁSTICOS	24
2.4.1 PROCESSOS ESTACIONÁRIOS	24
2.5 ESTADOS LIMITE	25
2.6 CONFIABILIDADE	26
2.6.1 SIMULAÇÃO DE MONTE CARLO	27
3 KRIGAGEM	29
3.1 KRIGAGEM SIMPLES	29
3.2 KRIGAGEM ORDINÁRIA	30
3.2.1 PROPRIEDADES DA KRIGAGEM ORDINÁRIA	32
4 METODOLOGIA	35
4.1 FERRAMENTAS USADAS PARA KRIGAGEM	35
4.2 HISTÓRICO DO DESENVOLVIMENTO DO PROGRAMA	36
4.2.1 PROGRAMA AK-MCS	37
4.2.2 DESCRIÇÃO DO PROGRAMA	39
4.3 FUNÇÃO DE APRENDIZADO	41

4.4 CARACTERIZAÇÃO DOS PROBLEMAS ESTUDADOS	42
4.4.1 EXEMPLO 1	42
4.4.2 EXEMPLO 2	42
4.4.3 EXEMPLO 3	43
4.4.4 EXEMPLO 4	44
5 RESULTADOS	47
5.1 ANÁLISE GERAL DO PROGRAMA	47
5.1.1 CASO 1 - NOÇÃO GERAL	47
5.1.2 CASO 2 - EFEITO DO NÚMERO DE PONTOS POR CICLO . . .	51
5.1.3 CASO 3 - EFEITO DO VALOR DE APRENDIZAGEM	54
5.2 ANÁLISE DE CADA EXEMPLO	57
5.2.1 EXEMPLO 1	57
5.2.2 EXEMPLO 2	59
5.2.3 EXEMPLO 3	60
5.2.4 EXEMPLO 4	62
 6 CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS	 65
 REFERÊNCIAS	 67

1 INTRODUÇÃO

Para fazer cálculos com incertezas pode-se utilizar por exemplo de ações e carregamentos variáveis, como aqueles relacionados ao vento, decaimento da resistência ou variações dimensionais, sendo cada um deles passível de ser escrito por variáveis aleatórias com um determinado comportamento. Portanto, é possível calcular e "manipular" os dados para determinar a confiabilidade de uma estrutura. Pode-se definir a probabilidade de falha ou a de sobrevivência de uma estrutura como um indicativo da confiabilidade de um dado sistema estrutural. Para obter a confiabilidade de um determinado projeto, deve-se utilizar de métodos de avaliação de confiabilidade, um deles é a simulação de Monte Carlo, usada neste trabalho. É importante ressaltar que neste trabalho será utilizada a simulação de Monte Carlo simples, porém existem outras.

O funcionamento da simulação de Monte Carlo consiste no uso de uma grande quantidade de pontos amostrais, sendo a quantidade necessária inversamente proporcional à probabilidade de falha que se deseja estimar. Sabendo que as estruturas em engenharia devem possuir uma segurança adequada, a quantidade de pontos para realização da simulação é elevada, sendo assim o custo computacional para realizar esta simulação é elevado.

De acordo com BECK(1), o custo computacional para a simulação de Monte Carlo é um limitante para alguns projetos. Portanto, para reduzir este custo computacional, elaborou-se um método de simulação de Monte Carlo utilizando a meta modelagem, sendo assim pode-se aumentar a utilidade da simulação de Monte Carlo. Para este estudo foi escolhido o modelo de krigagem, cujo funcionamento, uso e resultados serão apresentados ao longo do texto juntamente com o funcionamento da simulação que o utiliza.

1.1 JUSTIFICATIVA

Diversos problemas de engenharia encontram na complexidade e custo computacional uma barreira difícil de transpor, para confiabilidade não é diferente. Para o método de Monte Carlo o custo computacional depende do problema em questão, sabendo que deve-se ter um número grande de pontos amostrais para estimar a probabilidade de falha adequadamente pode-se afirmar que problemas de alta complexidade são impossíveis ou inviáveis para simulação.

Será mostrado neste trabalho que o número de pontos amostrais utilizados tende a ser por volta de 100 mil a 1 milhão, cada um desses pontos deve ser avaliado e calculado de acordo com o problema proposto. Os exemplos mostrados neste trabalho não tem um custo computacional que impossibilite esta simulação, porém imaginando que para um

projeto de edifício cada cálculo estrutural pode demorar dias a simulação demoraria anos, impossibilitando seu uso. Este estudo se originou com o objetivo de reduzir esse tempo e o custo computacional envolvido.

1.2 OBJETIVOS

O objetivo geral deste trabalho é verificar a possibilidade do uso de meta modelagem, mais especificamente krigagem, para problemas de confiabilidade por meio da simulação de Monte Carlo.

1.2.1 OBJETIVOS ESPECÍFICOS

Os objetivos específicos consistem em:

1. Utilizar, aprofundar e demonstrar o conhecimento na área de confiabilidade obtido durante a graduação;
2. Desenvolver uma ferramenta computacional para auxílio em problemas de confiabilidade;
3. Reduzir o custo computacional da simulação de Monte Carlo;
4. Observar possíveis soluções para os problemas encontrados.

1.3 ESTRUTURA DO TRABALHO

Este trabalho está estruturado com o objetivo de ser utilizado para estudos futuros, sendo iniciado com os conceitos básicos necessários, dando uma visão geral sobre os temas envolvidos, seguido de um capítulo sobre o modelo de krigagem, considerando a grande importância deste assunto para o estudo realizado. Para então serem apresentados a metodologia utilizada e os resultados obtidos e por fim serão feitas as devidas conclusões para o estudo realizado.

2 CONCEITOS BÁSICOS

Antes de ser possível analisar qualquer parte do trabalho, será necessário compreender os conceitos básicos em diversas áreas envolvidas durante os estudos. Primeiramente será explicado sobre variáveis aleatórias, teoria da probabilidade, processos estocásticos, confiabilidade e finalmente o método de Monte Carlo.

Todos os assuntos comentados serão explicados de forma apenas suficiente para a compreensão deste estudo, para aprofundar os conhecimentos sobre tais assuntos pode-se buscar os trabalhos de CHOI, GRANDHI e CANFIELD; PAPOULIS e PILLAI; SUDRET(2, 3, 4), além dos autores citados no decorrer do texto.

2.1 ESTATÍSTICA E PROBABILIDADE

De acordo com MONTGOMERY, RUNGER e HUBELE; SILVA(5, 6), a medida de uma determinada variável, como uma temperatura em um dado instante, é chamada de experimento. Frequentemente existirão fatores que influenciam esta medida que não são possíveis ter controle, seja em função do alto custo que teria para ser controlado ou em função da inexistência de um modo de anular este fator. Estes fatores são chamados de componentes aleatórios.

Um experimento que possa ser repetido da mesma maneira, porém que resulta um valor diferente é chamado de experimento aleatório. Para REDDY(7) o experimento aleatório é um dado experimento cujo seu resultado não pode ser exatamente previsto com antecedência. Um experimento ou medida aleatória é usualmente descrito como sendo uma variável aleatória. Para um determinado experimento, o conjunto de todos os possíveis valores medidos é denominado como espaço de probabilidades do experimento, chamado geralmente de "espaço amostral" e representado pela letra grega omega maiúsculo (Ω). Quando este espaço amostral é definido por pontos, a variável aleatória associada a ele é chamada de discreta, enquanto um espaço amostral definido por intervalos possui como origem uma variável contínua. O resultado de uma jogada de dados é discreto, enquanto uma medida de velocidade é uma variável contínua.

Os objetivos dos estudos de estatística e probabilidade são compreender, quantificar e modelar as variações dos objetos em estudo. É possível dizer que reduzir a variabilidade dos possíveis resultados seja um dos objetivos destes estudos também, porém não foi encontrado nenhum autor que fez esta afirmação, talvez por considerar o controle como sendo um assunto à parte, que usa a estatística como ferramenta.

2.2 TEORIA DA PROBABILIDADE

Inicialmente deve-se definir de maneira clara alguns termos usados na teoria de probabilidade, sendo estes:

- 1 - Resultado: é o resultado de uma única tentativa da variável ou processo em questão;
- 2 - Espaço Amostral (Ω): é a junção de todos os possíveis resultados para uma única tentativa. O espaço amostral da jogada de um dado comum por exemplo é: $\Omega = \{1;2;3;4;5;6\}$. O espaço amostral também é chamado de "Universo";
- 3 - Evento: é um grupo ou coleção de resultados, sendo estes reunidos de uma maneira específica. Para a jogada de dados poderia ser o evento "cair um número par" $E_{par} = \{2;4;6\}$;
- 4 - Complementar do evento: pode ser definido como sendo todos os resultados que não estão dentro do evento determinado, ele é geralmente denotado como A' , \bar{A} ou A^C . No exemplo acima temos $E_{par}^C = \{1;3;5\}$ para o complemento do evento "cair um número par". Neste trabalho será usada a notação A^C , pois as outras duas notações possuem sentido diferente nas teorias de conjuntos.

De acordo com BECK(1) existem alguns modos de definir a probabilidade. Considerando a ocorrência de um determinado evento A, tais definições são apresentadas a seguir.

2.2.1 DEFINIÇÃO CLÁSSICA

$$P[A] = \frac{N_A}{N} \quad (2.1)$$

Sendo N_A o número de vezes que um resultado do evento A ocorre dentro de uma amostra de tamanho N , considerando que temos a mesma chance de ocorrência entre todos os resultados da amostra (que são chamados de resultados "equi-prováveis").

2.2.2 DEFINIÇÃO FREQUENTISTA

$$P[A] = \lim_{n \rightarrow \infty} \frac{n_A}{n} \quad (2.2)$$

Sendo n_A o número de vezes que o evento A ocorre dentro de n tentativas. Esta definição baseia-se na realização de resultados e na observação, sendo muito usada para estimar a probabilidade em eventos do mundo real, porém possui uma limitação quanto ao número n de observações, uma vez que nunca chegará ao infinito.

2.2.3 DEFINIÇÃO BAYESIANA

A definição Bayesiana é amplamente conhecida como "grau de confiança", nesta definição a probabilidade de ocorrência de um evento A é definida pelo grau de confiança de um sujeito em relação a ocorrência deste evento. Alguns exemplos são: "Se eu jogar uma moeda tenho 50% de chance de tirar cara", "Ele deve ter uma chance em três de agarrar esse pênalti", "Acho que vai chover amanhã".

2.2.4 DEFINIÇÃO AXIOMÁTICA

Por não haver concordância entre os conceitos apresentados anteriormente, o matemático russo Andrei Nikolaievitch Kolmogorov descreveu a seguinte definição:

Em um experimento aleatório com um espaço amostral definido, uma função, representada por $P[A]$, que associa a cada resultado um número real é uma função de probabilidade se satisfaz as seguintes propriedades:

- i - Para qualquer evento A , $0 \leq P[A] \leq 1$;
- ii - A chance de ocorrer um resultado dentro do espaço amostral é 100%. Em outras palavras, um evento certo sempre ocorre. $P[\Omega] = 1 = 100\%$;
- iii - $P[A \cup B] = P[A] + P[B]$, desde que A e B sejam eventos incompatíveis, mutuamente excludentes no espaço amostral, ou seja, $[A \cap B \cup \Omega] = \emptyset$.

Estas propriedades não definem como a função $P[A]$ é construída, sendo usadas as outras definições ou outros métodos de adequação de experimentação para funções de probabilidade.

2.2.5 FUNÇÕES DE PROBABILIDADE E DENSIDADE DE PROBABILIDADE

A função de probabilidade $P[A]$, define a chance de um determinado evento A ocorrer. Utilizando do método de derivação do cálculo é possível definir a função $f_X(x)$ para associar a cada valor de x um valor de "densidade de probabilidade". Esta "densidade" não associa a um resultado x uma probabilidade, porém ao integrar esta função em um intervalo qualquer, é possível ter a probabilidade deste intervalo ocorrer.

Apesar de ser contra intuitivo o uso desta função de densidade, imagine a situação de uma medida de temperatura, ela pode ter infinitos valores possíveis entre 20 a 25 graus centígrados por exemplo. Para saber a probabilidade de ocorrer algum valor, consideramos de maneira lógica intervalos, como de 1 em 1 grau. Porém pode ser necessário alterar a faixa adotada, como de 0,5 em 0,5 graus, sendo assim, o modo mais fácil de descrever a função para gerar a probabilidade para qualquer intervalo é por meio da densidade de probabilidade, basta fazer uma integração entre quaisquer valores desejados.

Muitos dos eventos relevantes, em particular na área de engenharia, são definidos por um determinado valor limite x_0 dentro do seu espaço amostral. Supondo uma variável

aleatória contínua X definida no intervalo de $-\infty$ a $+\infty$, e que seja necessário estimar a probabilidade de a mesma assumir valores abaixo de um determinado valor x_0 , basta saber o valor da integral até o ponto x_0 , definido por:

$$P[X \leq x_0] = \int_{-\infty}^{x_0} f_X(x) dx \quad (2.3)$$

A notação $P[X \leq x_0]$ é chamada também de função acumulada até x_0 . Considerando a importância desta probabilidade acumulada até um valor limite x_0 e com o objetivo de manter a coerência com a notação do cálculo, denota-se usualmente como $F_X(x_0)$ a função de probabilidade acumulada avaliada em um x_0 qualquer.

2.2.6 CONCEITOS DAS VARIÁVEIS ALEATÓRIAS

Apesar de ter sido transmitida uma visão geral sobre variáveis aleatórias, é necessário para a utilização das mesmas a introdução a conceitos da teoria de probabilidade. Para aprofundamento destes conceitos pode-se utilizar do trabalho de muitos autores de estatística ou confiabilidade como por exemplo o trabalho de ROSS(8), que é usado como base na definição destes conceitos.

Considerando duas variáveis aleatórias X e Y :

1. Valor esperado:

Define-se a valor esperado ou esperança para uma variável como sendo o valor mais provável para a mesma, denotando como $E[X]$. Para uma variável com um número contável de possibilidades calcula-se como:

$$E[X] = \sum_{i=1}^n x_i P(x_i) \quad (2.4)$$

Sendo $P(x_i)$ a probabilidade do resultado de X ser x_i . Para uma variável contínua, utiliza-se da sua função de densidade de probabilidade:

$$E[X] = \int_{-\infty}^{+\infty} x f_X(x) dx \quad (2.5)$$

Em função tanto de sua utilidade quanto funcionalidade, o valor esperado é chamado de valor médio, usualmente representado por μ ou μ_X .

2. Desvio padrão e variância:

O desvio padrão e a variância podem ser definidos como uma medida de dispersão dos valores em relação à média e são representado respectivamente por σ e $Var[X]$. Ambos são amplamente utilizados, sendo importantes para caracterizar as variáveis aleatórias e poder compará-las, quanto ao grau de dispersão da variável. A variância é calculada de acordo com a equação abaixo, e o desvio padrão é a raiz quadrada da

variância.

$$\begin{aligned} Var[X] &= E[(X - \mu_X)^2] = E[X^2] - E[X]^2 = \\ &= \int_{-\infty}^{+\infty} (x - \mu_X)^2 f_X(x) dx = \sigma^2 \end{aligned} \quad (2.6)$$

3. Covariância:

A covariância, também chamada de variância conjunta, é uma medida de correlação entre duas variáveis aleatórias. Anteriormente comentou-se de duas variáveis aleatórias independentes, a medida de independência entre X e Y seria quantificada por meio da covariância, quanto maior a covariância, mais dependentes são as duas variáveis. A covariância pode ser calculada por meio da equação:

$$Cov[X, Y] = E[(X - \mu_X)(Y - \mu_Y)] = E[XY] - E[X]E[Y] \quad (2.7)$$

Com esta definição pode-se utilizar de métodos de junção e manipulação de variáveis aleatórias.

Para compreensão do método da krigagem, será explicado o efeito da multiplicação por um escalar e como somar duas variáveis aleatórias. Sejam as variáveis aleatórias X e Y , supondo que saibam suas características. É fácil demonstrar que o efeito da multiplicação de um escalar λ utilizando os conceitos apresentados anteriormente resulta em:

$$\begin{aligned} E[\lambda X] &= \lambda E[X] = \lambda \mu_X = \mu_{\lambda X} \\ Var[\lambda X] &= E[(\lambda X - \mu_{\lambda X})^2] = E[(\lambda X - \lambda \mu_X)^2] = \\ &= E[\lambda^2 (X - \mu_X)^2] = \lambda^2 Var[X] \end{aligned} \quad (2.8)$$

Para a soma de duas variáveis aleatórias pode-se encontrar na literatura os seguintes resultados:

$$\begin{aligned} E[X + Y] &= E[X] + E[Y] \\ Var[X + Y] &= \sigma_{X+Y}^2 = Var[X] + Var[Y] - 2Cov[X, Y] \end{aligned} \quad (2.9)$$

2.3 DISTRIBUIÇÕES RELEVANTES

Para este trabalho serão utilizadas algumas distribuições distintas, são elas a distribuição uniforme, normal, lognormal e distribuição de Gumbel. Estas distribuições serão brevemente explicadas a seguir.

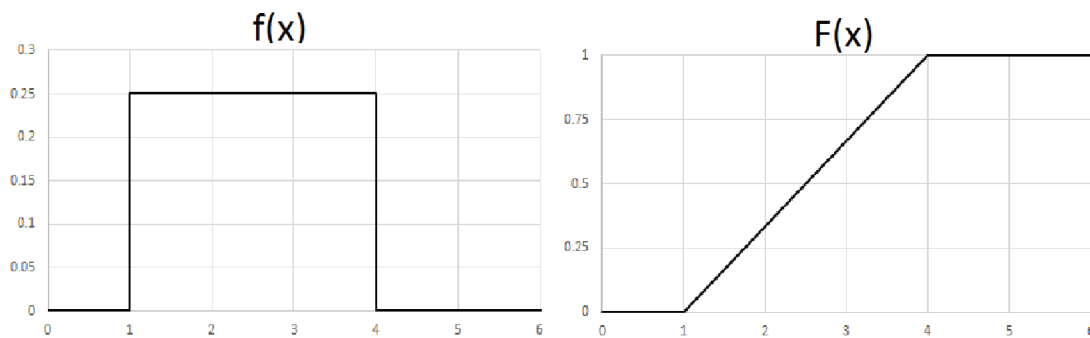
2.3.1 DISTRIBUIÇÃO UNIFORME

A distribuição uniforme consiste de uma função que associa a cada evento do espaço amostral uma chance proporcional apenas ao seu tamanho. Para dois eventos de mesmo

tamanho, a probabilidade de acontecer um é igual a probabilidade de acontecer o outro. A distribuição normal é uma das mais relevantes para programação, pois os geradores aleatórios de computador usualmente utilizam desta para gerar as outras distribuições. Para descrever esta distribuição, basta definir seu limite inferior e superior.

A figura 1 mostra a função de densidade de probabilidade e probabilidade acumulada respectivamente.

Figura 1 – Distribuição uniforme

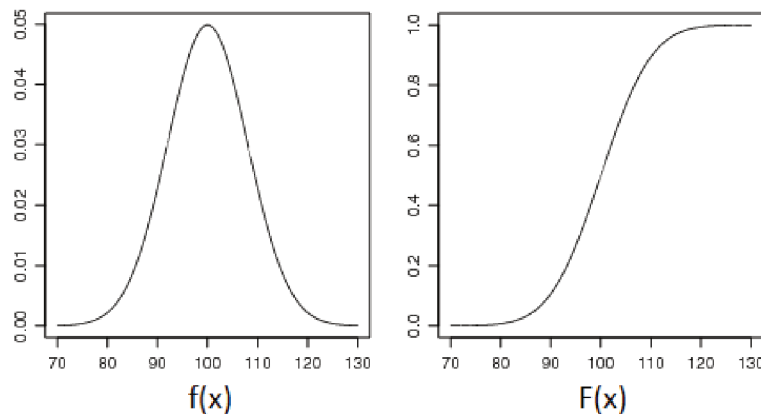


2.3.2 DISTRIBUIÇÃO NORMAL

A distribuição normal, também chamada de Gaussiana ou distribuição de Gauss-Moivre-Laplace, possui grande relevância nas áreas científicas, sendo a mais importante delas a distribuição normal padrão. Pode-se descrever a distribuição normal por sua média (μ) e seu desvio padrão (σ) [$X \sim N(\mu, \sigma)$], no caso da distribuição normal padrão, a média é igual a zero e o desvio padrão é 1. Por sua importância, a distribuição normal padrão possui notação própria para sua função de probabilidade acumulada, sendo esta Φ .

A figura 2 demonstra um exemplo de distribuição normal, sendo respectivamente sua função de densidade de probabilidade e sua probabilidade acumulada.

Figura 2 – Distribuição normal

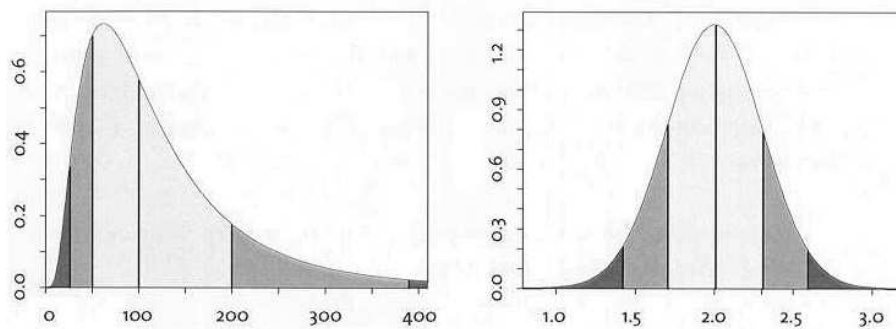


2.3.3 DISTRIBUIÇÃO LOGNORMAL

A distribuição lognormal pode ser descrita como a distribuição que, ao extrair seu logaritmo natural, resulta na distribuição normal. Ela pode ser descrita em função de sua média (λ) e desvio padrão (ζ) do logaritmo natural. Logo, se uma variável X é definida por uma distribuição lognormal, então: $X \sim LN(\lambda, \zeta) \iff \ln(X) \sim N(\lambda, \zeta)$.

A figura 3 faz uma comparação entre a distribuição lognormal e normal, sendo que as regiões com mesmo tom representam a mesma probabilidade de ocorrência.

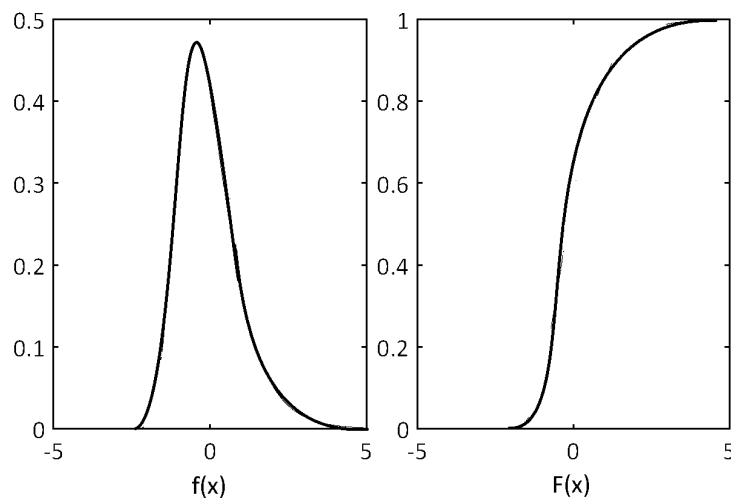
Figura 3 – Distribuição lognormal



2.3.4 DISTRIBUIÇÃO DE GUMBEL

A distribuição de Gumbel também é chamada de distribuição de valor extremo. Em algumas literaturas, a distribuição de Gumbel é usada para definir tanto valores extremos de máximo como de mínimo, neste estudo será utilizado sempre referindo-se à distribuição de Gumbel de valor extremo máximo. Pode ser denotada por $X \sim G(\mu, \beta)$.

Figura 4 – Distribuição de Gumbel



2.4 PROCESSOS ESTOCÁSTICOS

Dentro da teoria das probabilidades, um processo estocástico é uma família de variáveis aleatórias representando a evolução de um sistema de valores ao longo de uma variável contínua (usualmente o tempo), também chamado de processo aleatório, sendo este oposto ao processo determinístico. Em um processo estocástico há uma indeterminação, mesmo que se conheça a condição inicial, existem várias, senão infinitas, possibilidades de resultados.

Uma possível abordagem para descrever o processo estocástico seria considerá-lo como uma função de variáveis aleatórias no tempo. Para uma definição mais formal, um processo aleatório é definido por uma função $S(w, t)$, onde o w representa um ponto amostral de uma variável aleatória ao qual foi associada uma função em t . Pode-se analisar um processo estocástico fixando o w , tornando $S_w(t)$ uma função dependente de t , enquanto para um t fixo, obtêm-se $S_t(w)$ que seria uma variável aleatória no contínuo t , e finalmente fixando ambos será obtido $S(w, t)$ que resultará em um valor numérico definido pela realização da variável aleatória $S_t(w)$ e do resultado da função $S_w(t)$, que são iguais.

Portanto, o tratamento estatístico para o processo estocástico, apesar de ser diferente do usado para variáveis aleatórias, possui semelhanças. Neste trabalho, será usada a definição de processo estocástico utilizando espaços dimensionais, sendo definido posteriormente a função $Z_w(x)$.

2.4.1 PROCESSOS ESTACIONÁRIOS

Usualmente o termo "estacionário" é utilizado para processos estocásticos. Um processo é considerado estacionário quando não possui variação com o tempo, portanto seria uma variável aleatória simples, caso contrário ele é chamado de não estacionário. Pode-se definir um processo como fracamente estacionário, ou estacionário de segunda ordem se ele tem como propriedade a média, variância e covariância constantes no decorrer do tempo.

É possível estender o conceito de estacionário para campos aleatórios, sendo assim o conceito seria relacionado com a topologia. Pode-se descrever matematicamente um campo estacionário de segunda ordem por: dada uma função $G(x)$ de média m , diz-se que ela é fracamente estacionária se no campo aleatório gerado valem:

$$\begin{aligned} E[G(x)] &= m \\ E[G(x)^2] &< \infty \\ E[(G(x) - m)(G(x + h) - m)] &= E[G(x)G(x + h)] - m^2 \\ &= Cov(x, x + h) = Cov(h) \end{aligned} \tag{2.10}$$

Neste sentido, $E[\]$ representa a "esperança" da expressão interior dos colchetes,

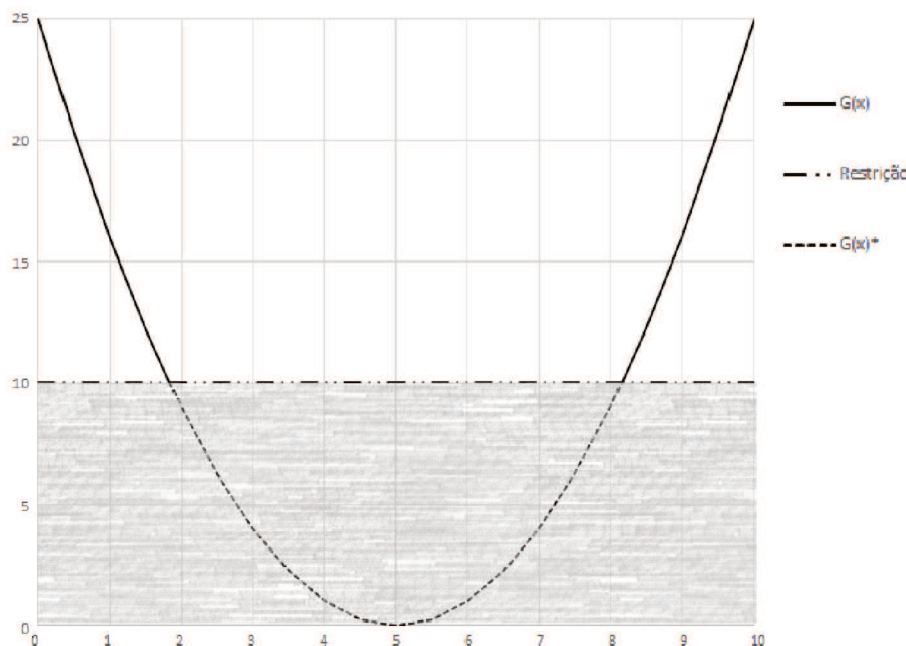
também chamado de valor esperado. Para o caso da função aleatória $G(x)$, a esperança é o valor médio m . A expressão $Cov(x_1, x_2)$ indica covariância entre x_1 e x_2 , que para ser uma função aleatória estacionária de segunda ordem, deve depender apenas da distância entre os dois pontos definida pela topologia.

2.5 ESTADOS LIMITE

Estados limite em engenharia são associados a critérios, condições ou situações de falha. Tais limites podem ser quantitativos ou qualitativos, um critério de falha quantitativo poderia ser associado a limpeza de uma cozinha por exemplo, ainda que seja possível criar critérios, sempre existirá uma certa subjetividade. Para este estudo, não será utilizado nenhum tipo de critério qualitativo, apenas quantitativos.

Sempre que um critério pode ser quantificado, é possível descrevê-lo como uma função de estado limite ou função limite. Esta função deve ser limitada por meio de uma imposição de restrição. Um exemplo simples é $G(x) = (x + 5)^2$ com a imposição de que resultados menores que 10 implicam em uma falha, este exemplo é ilustrado na figura 5 onde a linha contínua e tracejada representa a $G(x)$, a linha de ponto traço representa o limite imposto, sendo a região hachurada chamada de região de falha. Portanto o tracejado implica falha segundo $G(x)$ e a contínua indica a sobrevivência segundo o critério representado por $G(x)$.

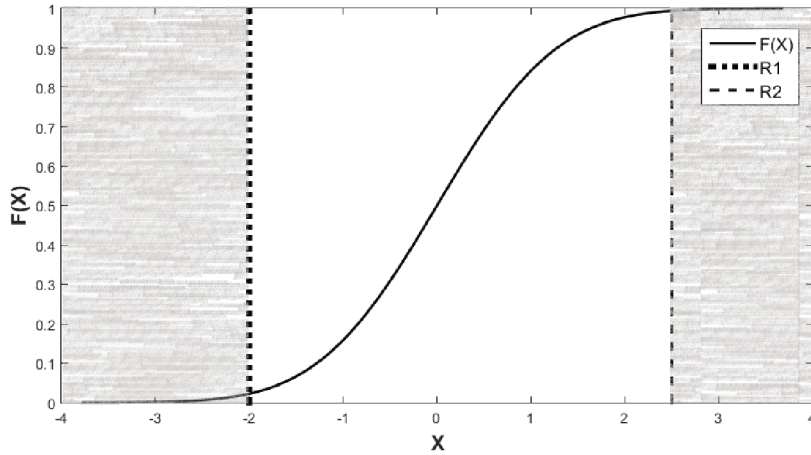
Figura 5 – Exemplo 1 de limite de falha



Apesar do caso acima ser uma função simples, pode-se utilizar de restrições complexas ou mais de uma restrição. Para exemplificar isto, será utilizada uma variável normal

padrão com duas restrições, a primeira, R_1 , implica em falha para $X \leq -2$, enquanto a segunda, R_2 , implica em falha para $X > 2,5$. Portanto a falha pode ser escrita como um evento E_f onde $E_f = \{X \leq -2\} \cup \{X > 2,5\}$. Para calcular a probabilidade de falha ($P[E_f]$) para este caso, basta utilizar a função de probabilidade de falha acumulada como mostra a figura 6, sendo as linhas conforme indicado na legenda e as partes hachuradas indicam os trechos correspondente à falha. Portanto, o cálculo da probabilidade de falha seria o valor de $F(x)$ para x igual a -2, que representa a probabilidade de x valer -2 ou menos, mais $1 - F(x)$ para x igual a 2,5, que representa a probabilidade complementar de x ser 2,5 ou menor, portanto maior que 2,5. Como a variável estudada é uma distribuição normal padrão pode escrever como: $P_f = P[E_f] = F(-2) + (1 - F(2,5)) = \Phi(-2) + (1 - \Phi(2,5))$.

Figura 6 – Exemplo 2 de limite de falha



Pode-se equacionar a probabilidade de falha para um determinado estado limite como:

$$P_f = \int_{D_f} f_X(x) dx \quad (2.11)$$

Onde P_f é a probabilidade de falha, D_f é o domínio de falha, ou seja, o intervalo de valores aos quais para a função limite resulte em falha e $f_X(x)$ é o valor da função de densidade de probabilidade avaliado no ponto x .

2.6 CONFIABILIDADE

A confiabilidade está relacionada à confiança que um sistema satisfaz a um determinado critério de falha, como foi demonstrado no exemplo 2 citado na explicação de estados limite. Pode-se mostrar seu valor por meio de porcentagens ou números, entretanto um meio amplamente utilizado de expressar a confiabilidade é representar por um índice de confiabilidade, denotado como β .

O valor de confiabilidade é expresso como sendo o número de desvios padrão que uma falha está distanciada do valor médio de uma determinada distribuição aleatória, então qualquer função aleatória pode ter sua confiabilidade expressa por β . Para o caso da distribuição normal padrão, que tem média zero e desvio padrão um, o número de desvios que um valor está distanciada da média em uma distribuição normal padrão é exatamente o valor absoluto deste número, logo igual a β . Para o exemplo citado anteriormente, onde foi utilizada a distribuição normal padrão a confiabilidade poderia ser descrita por $\beta_1 = |-2|$ para R_1 e $\beta_2 = |2, 5|$ para R_2 , para descrever em apenas um coeficiente de falha, deve-se calcular a probabilidade de falha $P_f = \Phi(-2) + (1 - \Phi(2, 5)) = 0,029 = 2,9\%$, e fazer o cálculo inverso $\Phi(-\beta) = 0,029 \rightarrow \beta \cong 1,9$.

2.6.1 SIMULAÇÃO DE MONTE CARLO

A simulação de Monte Carlo consiste de uma experimentação numérica extensiva. De acordo com BECK(1):

A simulação é uma técnica que permite a solução de problemas muito complexos. Tem sido utilizada para prever o comportamento a longo prazo de sistemas complexos de qualquer natureza. Na simulação, não há limite no número de variáveis do problema ou na complexidade de modelo, resolvendo-se com a mesma facilidade problemas com poucas ou muitas variáveis. O único fator limitante é a capacidade computacional, que tem aumentado muito nos últimos anos.

Pode-se definir a simulação de Monte Carlo pela seguinte definição:
Seja E_f o conjunto onde, para todo $x_0 \in E_f$ o sistema resulta em uma falha e P_f a probabilidade de falha do sistema. Então:

$$I[x] = \begin{cases} 1, & \text{se } x \in E_f \\ 0, & \text{se } x \notin E_f \end{cases} \quad (2.12)$$

$$P_f = \int_{\Omega} I[x] f_X(x) dx = E[I[x]] \quad (2.13)$$

Pela definição, é possível estimar o valor da integral por meio de uma amostra de tamanho finito por:

$$\widehat{P}_f = \frac{1}{n_{MC}} \sum_{i=1}^{n_{MC}} I[x_i] = \frac{n_f}{n_{MC}} \quad (2.14)$$

Na equação 2.14 está denotado \widehat{P}_f por ser uma estimativa de P_f , n_f indica o número de falhas que ocorreram dentro dos n_{MC} valores de x testados. Esta equação resulta numa estimativa não tendenciosa do valor real da probabilidade de falha, portanto para um $n_{MC} \rightarrow \infty$ o valor de \widehat{P}_f deve convergir para o valor exato.

Logicamente, quanto maior o número de experimentos, melhor a estimativa será, porém o custo computacional depende desse número, quanto maior o número de testes de x , maior o custo computacional e, de certa forma, o tempo de execução do problema. Diversas

fontes indicam que para ter um erro adequado deve-se considerar o desvio-padrão da probabilidade de falha estimada em relação com a probabilidade estimada, conhecida como coeficiente de variação e usualmente denotado por CV_{P_f} ou COV , do inglês *coefficient of variation*. Quanto menor o coeficiente de variação, melhor é a estimativa da probabilidade de falha em relação à real. Algumas vezes o coeficiente de variação é usado como informação sobre variáveis aleatórias, sabendo o valor médio (ou esperado) e o COV é possível descobrir a variância e desvio padrão.

$$COV = \frac{\sqrt{Var[\hat{P}_f]}}{E[\hat{P}_f]} \approx \frac{1}{\sqrt{n_{MC} P_f}} \quad (2.15)$$

Como dito anteriormente, o custo computacional é o limitante para a simulação de Monte Carlo. Em função disso, muitos estudos foram feitos para reduzir a variância da estimativa tentando reduzir o número de pontos estudados. Neste estudo em especial será demonstrada a alternativa desenvolvida envolvendo krigagem.

3 KRIGAGEM

As raízes do método de krigagem, de acordo com OLEA(9), vem dos estudos de Wold em 1938, Kolmogorov em 1939 e 1941 e Wiener em 1949, que não tiveram direta relação com os estudos de regressão e mineração de Krige em 1951, o qual foi popularizado na geo estatística por meio dos trabalhos de Matheron em 1962 e 1963. Entretanto, em LUENBERGER(10) foram feitos estudos objetivando problemas de otimização sem ênfase nas áreas de geo estatística, foi quando ele publicou uma teoria de otimização por meio de vetores do qual o sistema de equações de krigagem fazia parte.

O método de krigagem é uma forma de regressão linear para a formulação de um estimador. Sabe-se que atualmente existem diversas formas de krigagem, assim como existem outras alternativas de estimadores, mas a vantagem observada da krigagem seria o modo em que considera a influência de cada ponto em seu arredor. Além disso a curva de resultados estimada é exata para todos os pontos conhecidos, sendo assim, pode ser um bom modelo para estudos de confiabilidade como neste trabalho.

Os métodos de krigagem mais comuns atualmente são:

- 1 - Krigagem simples;
- 2 - Krigagem ordinária (usada neste trabalho);
- 3 - Krigagem global ou universal;
- 4 - Cokrigagem.

Neste trabalho será explicado com mais detalhes apenas a krigagem simples e ordinária, entretanto é interessante comentar um pouco sobre as diferenças mais gerais de cada. A simples considera que a média é constante e conhecida; a ordinária considera que a média é constante em cada região de estudo (para simplificação do modelo), mas não se sabe inicialmente o valor da mesma; a krigagem global considera que existe uma média única removendo as restrições do uso da krigagem ordinária; por fim, a cokrigagem é um método usado quando existe mais de uma variável de interesse.

Para uma abordagem mais aprofundada sobre o assunto recomenda-se os estudos de OLEA; JOURNAL e HUIJBREGTS; KROETZ; CORREIA(9, 11, 12, 13).

3.1 KRIGAGEM SIMPLES

Seja $Z(x)$ uma função de variável aleatória estacionária de segunda ordem na qual x representa um ponto. O estimador $\hat{Z}(x_0)$ no ponto x_0 é calculado com base na combinação linear dos pontos de controle x_i da seguinte forma:

$$\hat{Z}(x_0) = m + \sum_{i=1}^n [\lambda_i (Z(x_i) - m)] \quad (3.1)$$

Sendo cada λ_i o peso dado para cada x_i e n o número de pontos de controle. Para calcular os valores ideais para cada peso é preciso:

- 1 - Calcular a covariância de cada par de pontos (x_i, x_j) ;
- 2 - Calcular a covariância do ponto x_0 com cada um dos pontos conhecidos;
- 3 - Calcular cada λ_i de acordo com:

$$\begin{cases} \sum_{i=1}^n [\lambda_i \text{Cov}(x_i, x_1)] = \text{Cov}(x_0, x_1) \\ \sum_{i=1}^n [\lambda_i \text{Cov}(x_i, x_2)] = \text{Cov}(x_0, x_2) \\ \sum_{i=1}^n [\lambda_i \text{Cov}(x_i, x_3)] = \text{Cov}(x_0, x_3) \\ \dots\dots\dots \\ \sum_{i=1}^n [\lambda_i \text{Cov}(x_i, x_n)] = \text{Cov}(x_0, x_n) \end{cases} \quad (3.2)$$

- 4 - Computar o valor estimado para $\hat{Z}(x_0)$, sendo:

$$\hat{Z}(x_0) = m + \sum_{i=1}^n [Z(x_i) - m] \quad (3.3)$$

- 5 - E a estimativa da variância da krigagem por:

$$\sigma_{SK}^2 = \text{Cov}(0) - \sum_{i=1}^n [\lambda_i \text{Cov}(x_0, x_i)] \quad (3.4)$$

É importante notar que, no caso de existirem dois pontos x_i iguais, não existe solução única para o 3º passo.

3.2 KRIGAGEM ORDINÁRIA

Historicamente, a krigagem ordinária foi a primeira melhoria aplicada ao método de krigagem simples. Foi notado que os todos pesos da krigagem simples juntos não somavam 1, porém quando isso ocorria era possível remover o efeito do valor da média. Considerando o fato deste trabalho utilizar a krigagem ordinária, as definições e demonstrações serão apresentadas.

Assume-se que a amostra localizada nos pontos x_i resulte nas realizações da função aleatória $Z(x)$. Esta função aleatória deve ser contínua e estacionária de segunda ordem dentro do domínio das amostras. Tais afirmações permitem usar as propriedades elencadas nas equações 2.10, que serão reescritas aqui para facilitar o entendimento.

$$\begin{aligned} E[Z(x)] &= m \\ E[Z(x)^2] &< \infty \\ E[(Z(x) - m)(Z(x+h) - m)] &= E[Z(x)Z(x+h)] - m^2 \\ &= \text{Cov}(x, x+h) = \text{Cov}(h) \end{aligned} \quad (3.5)$$

O estimador criado por meio do método de krigagem ordinária com n pontos de controle é descrito como:

$$\begin{aligned}\hat{Z}(x_0) &= \sum_{i=1}^n \lambda_i Z(x_i) \\ \sum_{i=1}^n \lambda_i &= 1\end{aligned}\tag{3.6}$$

Para calcular cada peso λ_i ótimo, é necessário minimizar a variância da estimativa para o ponto x_0 descrito por:

$$\begin{aligned}\sigma_{KO}^2(x_0) &= Var[\hat{Z}(x_0) - Z(x_0)] \\ \sigma_{KO}^2(x_0) &= Var\left[\sum_{i=1}^n \lambda_i Z(x_i) - Z(x_0)\right]\end{aligned}\tag{3.7}$$

Definindo uma função $Y(x) = Z(x) - E[Z(x)] = Z(x) - m$ pode-se reescrever a equação 3.7 fazendo uma adição e subtração com valor m dentro da variância, lembrando que a soma de todos os pesos vale 1:

$$\begin{aligned}\sigma_{KO}^2(x_0) &= Var\left[\sum_{i=1}^n \lambda_i (Z(x_i) - m) - (Z(x_0) - m)\right] \\ \sigma_{KO}^2(x_0) &= Var\left[\sum_{i=1}^n \lambda_i Y(x_i) - Y(x_0)\right] \\ \sigma_{KO}^2(x_0) &= Var\left[\sum_{i=0}^n \lambda_i Y(x_i)\right]\end{aligned}\tag{3.8}$$

Sendo o último passo válido para $\lambda_0 = -1$. Para prosseguir deve-se mostrar que:

$$Var\left[\sum_{i=0}^n \lambda_i Y(x_i)\right] = \sum_{i=0}^n \sum_{j=0}^n \lambda_i \lambda_j Cov(Z(x_i), Z(x_j))\tag{3.9}$$

Pelo que foi exposto nas equações 2.6, segue que:

$$\begin{aligned}Var\left[\sum_{i=0}^n \lambda_i Y(x_i)\right] &= E\left[\left\{\sum_{i=0}^n \lambda_i Z(x_i)\right\}^2\right] - \left\{E\left[\sum_{i=0}^n \lambda_i Z(x_i)\right]\right\}^2 \\ Var\left[\sum_{i=0}^n \lambda_i Y(x_i)\right] &= \sum_{i=0}^n \sum_{j=0}^n \lambda_i \lambda_j E[Z(x_i)Z(x_j)] - \sum_{i=0}^n \sum_{j=0}^n \lambda_i \lambda_j E[Z(x_i)]E[Z(x_j)] \\ Cov_Z(x_i, x_j) &= E[(Z(x_i) - E[Z(x_i)])(Z(x_j) - E[Z(x_j)])] = E[(Z(x_i) - m)(Z(x_j) - m)] \\ E[(Y(x_i) - 0)(Y(x_j) - 0)] &= E[(Y(x_i) - E[Y(x_i)])(Y(x_j) - E[Y(x_j)])] = Cov_Y(x_i, x_j)\end{aligned}\tag{3.10}$$

Para terminar a prova da equação 3.10 basta reorganizar os termos da terceira linha, sendo a última parte da equação para provar que a covariância avaliada por meio de Y leva ao mesmo resultado que a covariância avaliada por meio de Z . Então utilizando das equações 3.8 e 3.10 pode-se chegar a:

$$\sigma_{KO}^2(x_0) = \sum_{i=0}^n \sum_{j=0}^n \lambda_i \lambda_j Cov(Z(x_i), Z(x_j))\tag{3.11}$$

Com mais um pouco de organização dos termos é possível chegar a:

$$\sigma_{KO}^2(x_0) = Cov(0) - 2 \sum_{i=1}^n \lambda_i Cov(x_i, x_0) + \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j Cov(x_i, x_j) \quad (3.12)$$

Portanto, de acordo com as equações 3.6 e 3.12, é necessário resolver um problema de otimização com restrição. Utilizando-se do método de Lagrange é possível simplificar esta otimização por uma sem restrições da seguinte maneira:

$$L(\lambda_1, \dots, \lambda_n, \mu_L) = \sigma_{KO}^2(x_0) + 2\mu_L \left(\sum_{i=1}^n \lambda_i - 1 \right) \quad (3.13)$$

Portanto o valor mínimo é atingido quando as derivadas parciais das incógnitas são nulas. É possível notar com essas funções o motivo de usar $2\mu_L$ como multiplicador de Lagrange, sendo as derivadas:

$$\begin{aligned} \frac{\partial L(\lambda_1, \dots, \lambda_n, \mu_L)}{\partial \lambda_i} &= 2\mu + 2Cov(x_i, x_0) - 2 \sum_{j=1}^n \lambda_j Cov(x_i, x_j); \forall i \in \{1, \dots, n\} \\ \frac{\partial L(\lambda_1, \dots, \lambda_n, \mu_L)}{\partial \mu_L} &= 2 \sum_{i=1}^n \lambda_i - 1 \end{aligned} \quad (3.14)$$

Podemos escrever então como um sistema de equações que resulta no valor ótimo dos pesos λ_i :

$$\begin{cases} Cov(x_1, x_0) = \sum_{j=1}^n \lambda_j Cov(x_1, x_j) - \mu \\ Cov(x_2, x_0) = \sum_{j=1}^n \lambda_j Cov(x_2, x_j) - \mu \\ \dots\dots\dots \\ Cov(x_n, x_0) = \sum_{j=1}^n \lambda_j Cov(x_n, x_j) - \mu \\ \sum_{i=1}^n \lambda_i = 1 \end{cases} \quad (3.15)$$

3.2.1 PROPRIEDADES DA KRIGAGEM ORDINÁRIA

De acordo com OLEA; JOURNAL e HUIJBREGTS(9, 11) a krigagem ordinária contém as seguintes propriedades:

- 1 - Possui o desvio padrão da krigagem minimizado;
- 2 - O intervalo de uso do estimador gerado não é restringido pelo intervalo das amostras;
- 3 - O estimador resulta em valores de peso maiores para amostras próximas do ponto estimado;
- 4 - A krigagem ordinária evita agrupamentos de pontos com alto peso, levando em consideração os pesos próximos;
- 5 - Os pontos de controle continuam exatos pelo método;
- 6 - Incapacidade de lidar com pontos de controle repetidos;
- 7 - Independência da origem do espaço amostral, leva em consideração apenas as distâncias relativas.

Estas propriedades permitem um bom desempenho nos valores estimados. As propriedades 1 e 3 tentam garantir uma boa estimativa para todos os pontos. A 2 indica que não é necessário encontrar valores máximos para o problema antes de utilizar o modelo de krigagem. A propriedade de número 4 tenta garantir estimativas não tendenciosas. A 5 indica que os valores conhecidos permanecem conhecidos, isso é útil para definir os pontos mais importantes como será mostrado. A propriedade de número 6 indica que o modelo de krigagem pode não servir para alguns tipos de problema ou acabar falhando caso seja mal desenvolvido. Por fim a número 7 indica que a origem do espaço amostral pode ser deslocada livremente sem alterar o resultado final do estimador.

4 METODOLOGIA

Após a apresentação dos conceitos e sobre o modelo de krigagem propriamente dito, pode-se finalmente apresentar como o trabalho foi desenvolvido e aprofundado. Para começar os estudos de krigagem, utilizou-se de um programa de computador disponível na literatura para entender como o método funcionava de maneira simplificada, sem a preocupação de aprender todos os assuntos envolvidos no primeiro momento. Após alguns resultados de testes, observou-se a utilidade e possível capacidade de uso para os problemas de confiabilidade. Com isso começou-se o estudo que está sendo apresentado aqui.

Primeiramente serão mostradas as ferramentas computacionais usadas, seguida da explicação detalhada do programa desenvolvido e finalmente serão mostrados os problemas resolvidos para a demonstração dos resultados.

4.1 FERRAMENTAS USADAS PARA KRIGAGEM

Como foi dito anteriormente, foi utilizado um pacote de ferramentas de terceiros para a execução da krigagem. Em primeiro momento foi utilizada a krigagem do "*DACE - Design and Analysis of Computer Experiments*" desenvolvido por Hans Bruun Nielsen, Soren Nymand Lophaven e Jacob Sondergaard. A documentação e o código do programa podem ser encontrados no site da universidade técnica da Dinamarca.

Apesar da utilidade das ferramentas desenvolvidas no DACE, foi encontrado um outro pacote de ferramentas para krigagem mais adequado para o estudo realizado, o pacote "*UqLab*" desenvolvido pelo grupo "*Risk, Safety and Uncertainty Quantification*" do instituto federal de tecnologia de Zurique, Suíça, com a supervisão do Prof. Dr. Bruno Sudret e do Dr. Stefano Marelli.

As ferramentas desenvolvidas no pacote *uqlab* são úteis tanto para o método de krigagem como confiabilidade e estatística. O funcionamento da krigagem neste pacote não pode ser facilmente descrito em detalhes, porém os manuais presentes no site <www.uqlab.com> podem ser usados para compreender cada uma das ferramentas disponíveis.

Cada manual é relacionado a um módulo, sendo usados neste trabalho os módulos de entradas, modelos e krigagem, considerando que para a utilização da krigagem todos estes foram necessários. Todos os manuais possuem duas partes, uma parte teórica e uma parte funcional. Na parte teórica é dada uma breve explicação da teoria necessária para compreender o módulo, enquanto na parte funcional é descrito o funcionamento da ferramenta envolvida, a descrição completa de cada entrada e saída necessária e as configurações passíveis de alteração por meio de comandos. Os manuais são bem escritos e devem ser de fácil uso para qualquer pessoa com noções de programação.

O módulo de entradas apresenta como descrever as variáveis aleatórias por meio do *uqlab*. Uma vez que este módulo é utilizado na geração das amostras aleatórias para a krigagem, optou-se por utilizá-lo também para validação do modelo usando a simulação de Monte Carlo simples com o objetivo de garantir que o gerador de aleatórios fosse o mesmo. Entretanto, é importante ressaltar que não foi identificada nenhuma diferença perceptível nos resultados entre o valor gerado pelo módulo de entradas ou pelo gerador nativo do *matlab*.

O módulo de modelos foi necessário para descrever a junção das variáveis aleatórias e avaliá-las de maneira matricial. A existência deste módulo foi um dos motivos de optar-se pelo uso do *uqlab*, considerando que pode-se ter um ganho de tempo em relação ao modelo do DACE. Pelos testes feitos, o *uqlab* foi em média de duas a três vezes mais rápido.

Por fim, o módulo de krigagem foi utilizado para resolver o sistema de equações 3.15. As configurações da resolução do sistema não foram alteradas, porém seria possível testar outros métodos de otimização com o objetivo de verificar se o ponto de ótimo para minimização do erro pode ser melhorado. Também evitou-se alterar qualquer configuração que não fosse necessária com o objetivo de facilitar a validação dos valores aqui apresentados.

4.2 HISTÓRICO DO DESENVOLVIMENTO DO PROGRAMA

O objetivo principal do programa desenvolvido era aproveitar o modelo de krigagem ao máximo para o cálculo de confiabilidade por simulação de Monte Carlo. O programa iniciou de uma maneira simples, gerando um campo de pontos de controle para criar o modelo e avaliando todos os pontos da simulação de Monte Carlo por meio da krigagem.

Após diversos testes, chegou-se à conclusão de que este método não daria resultados bons em função da natureza do problema. A simulação de Monte Carlo precisava de um grande número de pontos amostrais em função da probabilidade de falha, com probabilidades distintas de ocorrer a falha ou não. Testou-se a geração de pontos de controle por meio de padrões lineares, como uma malha de pontos quadriculada com cada quadrado possuindo um ponto de teste, porém este método gerava pontos excessivos para o modelo de krigagem, tornando o sistema de equações muito extenso, demorado e custoso para solucionar, ainda que pudesse ser melhor que a função original.

A seguir, tentou-se usar a geração dos pontos de controle com mesma distribuição aleatória dos pontos amostrais da simulação de Monte Carlo, porém, quando as probabilidades de falha eram baixas, os pontos de controle representavam a região (ou regiões) de falha de maneira insatisfatória, pois era improvável que os pontos de controle fossem sorteados próximos da falha. Novamente chegou-se no problema de ter muitos pontos para uma boa representação, gerando também um sistema de equações extenso e custoso.

Apesar de ambos os métodos anteriores terem resultados suficientemente aceitáveis,

era quase impossível detectar um modo de estimar o tamanho da amostra necessária, tornando improvável o uso deles em situações reais. Desenvolveu-se então uma versão com os pontos da simulação de Monte Carlo sorteados para serem pontos de controle da krigagem. Pelo mesmo motivo da tentativa anterior, persistiu o problema de pontos excessivos para conseguir bons resultados em função da baixa chance de cair próximo da região de falha.

O primeiro método funcional foi baseado na ideia da simulação de Monte Carlo com amostragem por importância, os pontos de controle eram gerados buscando a região de falha. Utilizou-se uma alteração nos valores de desvio padrão para a geração dos pontos de controle em relação aos que seriam utilizados no Monte Carlo. Supondo que a função de falha fosse algo como " $E_{falha}(x) = 1, se f(x) \leq 0$ ", sendo $f(\mathbf{X}) = X_1 - X_2$, com X_1 e X_2 variáveis aleatórias. Gerava-se os pontos de controle usando o desvio padrão dobrado ou triplicado para X_1 e X_2 . Com este procedimento era possível gerar pontos de controle mais espalhados e, considerando que a região de falha não é deslocada, a probabilidade de um ponto estar na região de falha era consideravelmente maior. Pensou-se nesta forma com o propósito de gerar os pontos de maneira similar ao Monte Carlo, porém distribuí-los melhor por todo o espaço amostral.

Apesar dos resultados dos estudos estarem indo conforme desejado, em função do tempo e outros interesses para o uso do modelo de krigagem, optou-se por testar e melhorar o procedimento denominado de AK-MCS dos estudos de ECHARD, GAYTON e LEMAIRE(14). Por fim, o procedimento estava dando bons resultados, mas havia necessidade de fazer alterações e modificações. Escolheu desenvolver este trabalho em cima deste processo e suas alterações para o estudo de krigagem em confiabilidade, considerando que suas conclusões poderão ser utilizadas para começar trabalhos futuros.

4.2.1 PROGRAMA AK-MCS

AK-MCS é uma sigla do inglês para *Active learning reliability method combining Kriging and Monte Carlo Simulation* ou apenas *Active Kriging for Monte Carlo Simulation*, ele consiste do uso de krigagem para avaliar os pontos amostrais, porém com uma função de aprendizado intermediando a necessidade de acrescentar novos pontos de controle para o sistema de equações da krigagem. O procedimento desenvolvido nos estudos de ECHARD, GAYTON e LEMAIRE(14) será descrito aqui, em função da importância do mesmo para este trabalho.

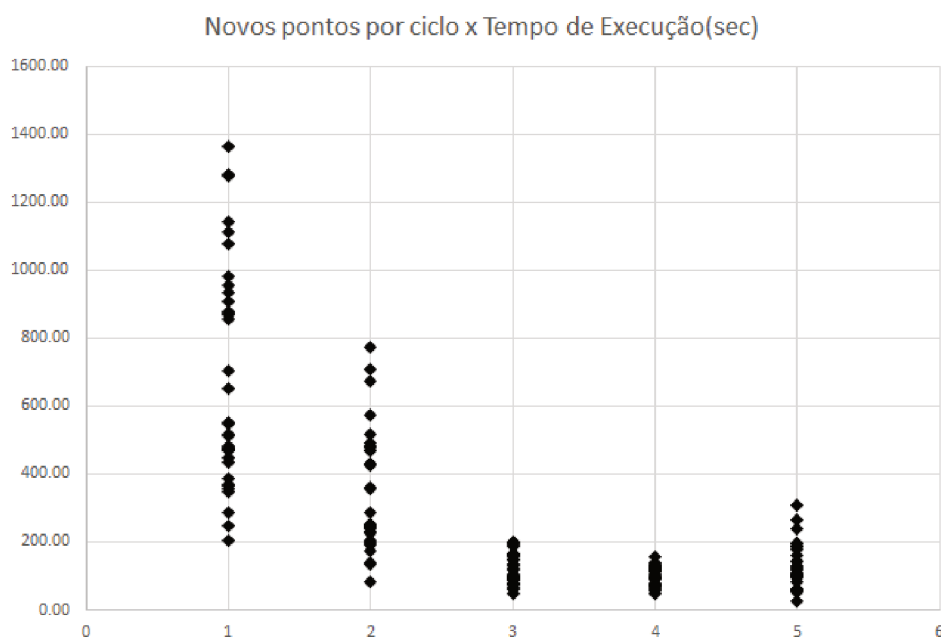
Inicialmente o AK-MCS gera os pontos amostrais da simulação de Monte Carlo e sorteia alguns pontos iniciais para montar o primeiro modelo de krigagem, sendo para o estudo citado utilizado o DACE para calcular o modelo de krigagem. Após a krigagem ter sido gerada, calculam-se todos os pontos amostrais de acordo com o modelo. Na sequência os pontos são avaliados de acordo com uma função de aprendizado e, caso não seja atendido

o critério de parada, o melhor ponto é escolhido para ser adicionado nas equações do modelo e volta-se para a etapa do cálculo do modelo de krigagem. Após atendido o critério de parada compara-se o COV da probabilidade de falha com 0,05. Caso o coeficiente de variação esteja elevado, aumenta-se o tamanho da amostra e o programa é reiniciado; caso seja menor, o AK-MCS mostra os resultados obtidos.

A principal alteração feita foi em relação aos pontos adicionados. Esperava-se ser capaz de testar alterações quanto a forma de sorteio dos pontos iniciais, testar a inclusão de pontos fora da amostra de Monte Carlo ao modelo de krigagem inicial e a melhoria da função de aprendizagem e critério de parada. Infelizmente o tempo limitou os estudos e não obteve-se bons resultados com outras alterações testadas, portanto foram desprezadas temporariamente, ainda pretende-se estudar tais alterações com o propósito de melhorar ainda mais o programa.

No geral, a alteração feita e apresentada neste trabalho foi capaz de reduzir o tempo de processamento do programa consideravelmente. Pode-se observar na figura 7 os tempos totais de execução para o exemplo 3, o tempo do programa AK-MCS original seria a primeira coluna de pontos, variando de aproximadamente 200 segundos até quase 1400 segundos, enquanto o tempo máximo com pelo menos 2 pontos novos adicionados no modelo de krigagem por ciclo resultou numa redução de tempo de 50% aproximadamente e para mais de 2 pontos adicionados por ciclo, o pior tempo de execução ficou menor que 350 segundos. Estes valores comparam todos os testes realizados para o exemplo 3, sendo mais de 200 testes variando a semente de geração aleatória, o número de pontos iniciais e o valor limite de aprendizado.

Figura 7 – Comparação de tempo de execução para exemplo 3



4.2.2 DESCRIÇÃO DO PROGRAMA

Para descrever o procedimento estudado será utilizada a Figura 8. Com ela podemos ver de maneira geral o funcionamento do programa. Para uma descrição mais detalhada, deve-se descrever cada etapa do programa individualmente, porém será necessário detalhar inicialmente os nomes de cada variável para a compreensão do programa:

Sample: é o nome dado para a amostra de Monte Carlo (poderia haver confusão com a amostra da krigagem);

Pontos de controle: é o nome dado para o conjunto de pontos que pertencem a amostra da krigagem;

nInit: é o número de pontos iniciais para a krigagem;

nFinal: é o número de pontos de controle do modelo de krigagem após o último ciclo;

β e P_f : de acordo com a notação de confiabilidade, índice de confiabilidade e probabilidade de falha, respectivamente;

SampleNum: O número total de pontos amostrais da simulação de Monte Carlo, inclui os pontos de controle;

Ciclo: Um ciclo é definido pelo número de vezes que o modelo de krigagem é feito e usado para avaliar toda a amostra, também é o número de vezes que verifica-se a função de aprendizagem para todos os pontos amostrais;

nAdd: Indica o número de pontos que são adicionados a cada ciclo do programa, notando que não são adicionados caso o programa caminhe para a finalização;

Função restringida: A função submetida a uma determinada restrição. Apesar de não ter sido feito, não deve haver problemas no uso de mais de uma restrição para a mesma função considerando o funcionamento do programa e seu equacionamento.

1° Passo: Gerar a amostra *Sample* usada para o Monte Carlo.

Como foi explicado anteriormente, a simulação de Monte Carlo utiliza de experimentação e deve-se ter uma amostra de tamanho adequado. Nesta etapa cada ponto amostral é gerado, mas não é calculado o valor da equação de estado limite a ele associado.

2° Passo: Sortear os pontos de controle iniciais.

Nesta etapa são escolhidos *nInit* pontos de controle, sendo eles sorteados dentro dos pontos da amostra.

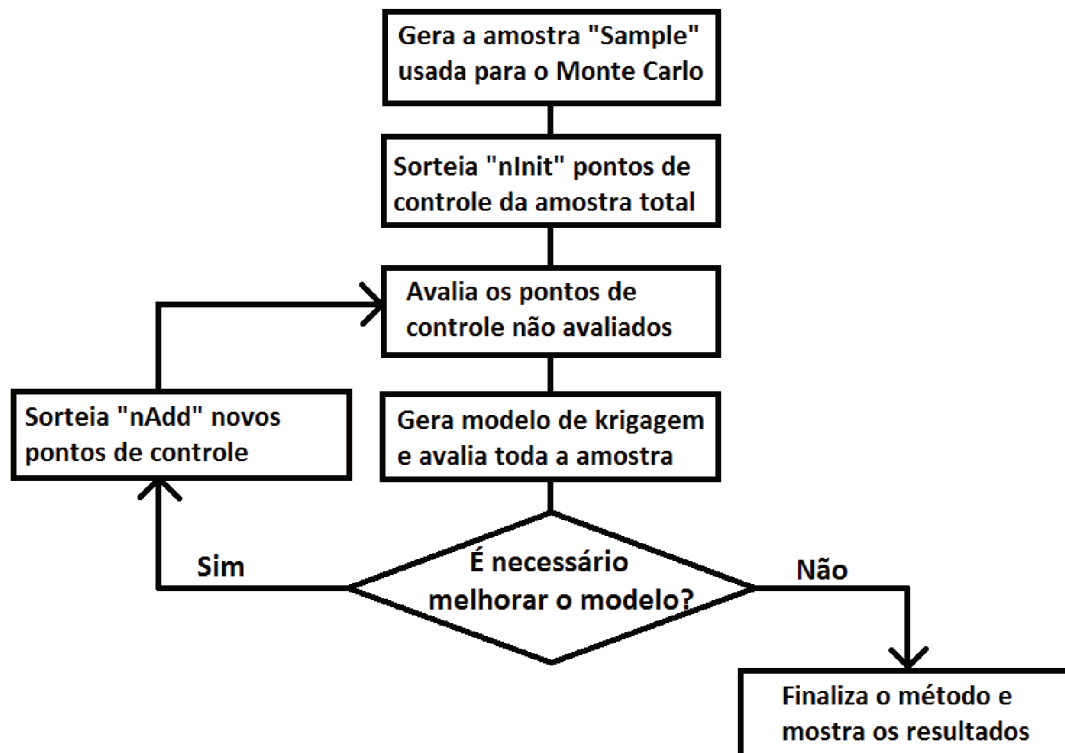
3° Passo: Avaliar os pontos de controle ainda não avaliados.

Os pontos de controle não avaliados, que inicialmente são todos os pontos sorteados, devem ser verificados na função restringida, sendo o valor obtido armazenado e atribuído ao valor estimado do erro um valor nulo, lembrando que a krigagem estima o resultado exato para pontos conhecidos.

4° Passo: Gera o modelo de krigagem e avalia a amostra.

Primeiramente os pontos de controle são utilizados para gerar os sistemas de equações

Figura 8 – Esquema do programa desenvolvido



da krigagem, a seguir este sistema é solucionado para todos os pontos da amostra total. Neste caso, são armazenados os valores da função e os desvios de erro de estimativa para cada ponto.

5º Passo: [Tomada de decisão] Checagem de aprendizagem.

Neste passo, avalia-se o valor de aprendizagem determinado por uma função de aprendizagem escolhida e compara-se o melhor candidato para novo ponto de controle com o valor limite de aprendizagem. Caso o melhor candidato para ponto de controle esteja dentro do limite de aprendizado, deve-se ir para o passo "I", senão ir para o passo 6.

É importante comentar que, para este passo foi incluída uma checagem adicional, caso nenhum ponto fosse estimado na situação de falha o programa ainda sim faria a adição de novos pontos de controle, sendo estes também escolhidos segundo o passo "I".

Passo I: Sorteia $nAdd$ novos pontos de controle.

O passo I é um passo intermediário, sendo escolhidos pontos adicionais para melhorar o modelo de krigagem. Neste caso, são adicionados os $nAdd$ pontos novos com melhor valor de aprendizagem. É importante notar que podem existir entre os novos pontos de controle alguns com valores de aprendizagem que não passariam no teste do 5º passo. Após esta etapa, volta-se para o 3º passo.

6º Passo: Finaliza o método e mostra os resultados.

Por fim, o programa verifica o número de pontos que violaram a restrição e mostra os

valores obtidos para P_f e β . O programa aceita entradas e saídas multidimensionais, no caso ele mostraria como resultados cada par de P_f e β .

Para este 6º passo, é importante ressaltar que em problemas de confiabilidade usuais deve-se comparar o tamanho da amostra com a probabilidade de falha e decidir se está adequada com base no coeficiente de variação (COV). Apesar do programa deste estudo não utilizar esta verificação, caso seja necessário, poderia se reiniciar o programa aumentando o tamanho da amostra de Monte Carlo.

4.3 FUNÇÃO DE APRENDIZADO

Foi comentado anteriormente sobre a função de aprendizado, porém não foi dito nada de maneira específica. A função de aprendizado pode ser qualquer função para ordenar os pontos amostrais, até mesmo um "desorganizador" poderia ser usado para determinar o valor de aprendizado, entretanto seria uma péssima forma de classificação.

Durante os trabalhos realizados, foram testadas algumas funções de aprendizado, algumas baseadas no valor da função restritiva, outras na distância entre pontos e até mesmo com critérios de aleatoriedade que poderiam fazer o resultado melhorar. Por fim decidiu-se classificar os pontos amostrais em função do desvio de erro de estimativa do modelo de krigagem e da proximidade do valor da função restritiva ao limite de restrição. Para generalizar a função de aprendizado para mais de um limite de restrição, recomenda-se usar como distância até o limite de restrição o menor valor dentre as distâncias, ou seja, o melhor aprendizado do ponto.

Ao longo da demonstração dos resultados os valores de aprendizado vão ser usados constantemente. Para facilitar a escrita (principalmente em tabelas com tamanho de texto limitado) será denominado U_i o valor de aprendizado do ponto i , U_{lim} o valor limite de aprendizado e no caso de surgir um U sem índice, será representando o melhor valor de aprendizado de um dado ciclo (o último quando não especificado).

A função que representa o valor de aprendizado para um ponto i é:

$$U_i = \frac{|\hat{Z}(x_i) - f_b|}{\sigma_{KO}(x_i)} \quad (4.1)$$

Sendo que $\hat{Z}(x_i)$ e $\sigma_{KO}(x_i)$ vêm diretamente do modelo de krigagem e f_b significa *function boundaries*, ou seja, o "limite" da função restritiva. Neste caso, quanto menor o valor de U_i , maior o aprendizado que ela tem a oferecer. Esta equação pode ser traduzida como: quanto mais próximo o valor estimado estiver do limite de falha, mais qualificado é o ponto, assim como quanto maior a incerteza do estimador sobre o ponto, melhor é seu valor de aprendizagem.

4.4 CARACTERIZAÇÃO DOS PROBLEMAS ESTUDADOS

Para este estudo, foi feita a análise de quatro problemas no total, dois deles são exemplos numéricos simples e dois deles envolvem o uso de elementos finitos em uma placa laminada de compósitos para obter a condição de falha. Com o objetivo de reduzir o tamanho das tabelas será usado o termo em inglês "*Seed*" para indicar o valor da semente de geração dos números aleatórios. Para os exemplos 3 e 4 foram utilizados programas de elementos finitos para gerar as equações de estado limite, tais programas foram desenvolvidos por LOPEZ *et al.*(17).

4.4.1 EXEMPLO 1

Este é o mais simples dos problemas estudados neste trabalho, considera-se duas variáveis normais padrão independentes com a seguinte função restritiva:

$$G(\mathbf{V}) = 5 - 0,5(V_1 - 0,1)^2 - V_2 \quad (4.2)$$

Sendo que valores menores ou iguais a zero de $G(\mathbf{V})$ indicam a falha do sistema. Para este problema a amostra utilizada possui 1000000 pontos, ou seja, 10^6 .

Tabela 1 – Valores para o método de Monte Carlo do exemplo 1.

Identificador	$P_f(\times 10^{-3})$	Beta
<i>Seed</i> 0	3,04	2,74
<i>Seed</i> 1	3,02	2,75
<i>Seed</i> 2	2,97	2,75
<i>Seed</i> 3	3,01	2,75
<i>Seed</i> 4	3,01	2,75
Média(<i>Seed</i> 0 a 29)	3,01	2,75
Maior P_f (0 a 29)	3,12	2,73
Menor P_f (0 a 29)	2,91	2,76

4.4.2 EXEMPLO 2

Este exemplo, ainda simples, considera três variáveis aleatórias independentes, com distribuições normais com médias 600, 1000 e 2 e desvios padrão de 30, 33 e 0,1, respectivamente. A restrição imposta ainda considera a falha apenas com valores iguais a zero ou menores e segue a seguinte função:

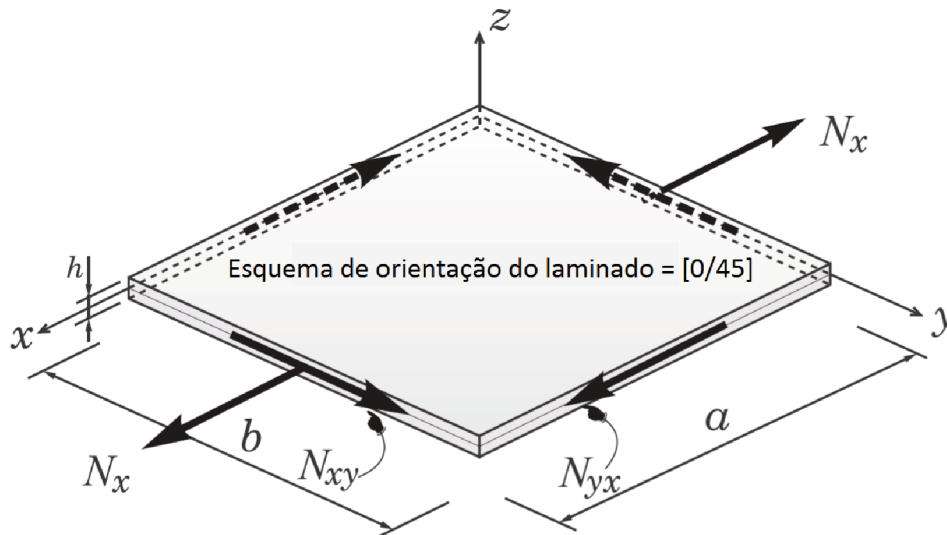
$$G(\mathbf{V}) = V_1 - \frac{V_2}{V_3} \quad (4.3)$$

Tabela 2 – Valores para o método de Monte Carlo do exemplo 2.

Identificador	$P_f(\times 10^{-2})$	Beta
<i>Seed</i> 0	1,20	2,26
<i>Seed</i> 1	1,18	2,26
Média(<i>Seed</i> 0 a 29)	1,21	2,25
Maior P_f (0 a 29)	1,24	2,24
Menor P_f (0 a 29)	1,19	2,26

4.4.3 EXEMPLO 3

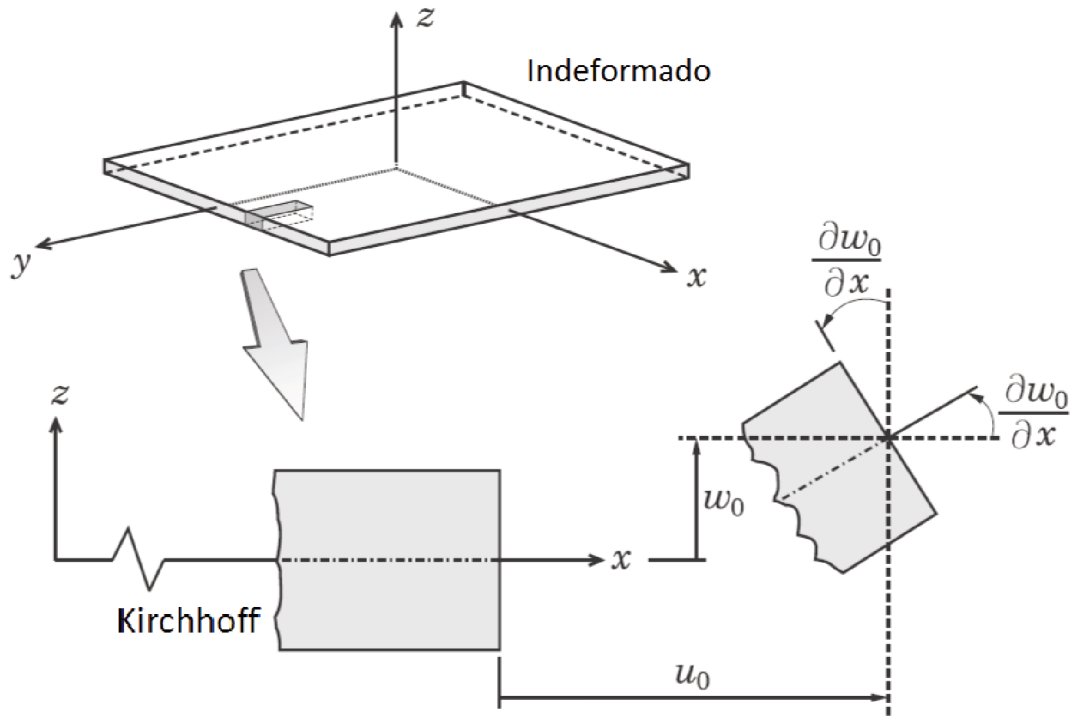
Este exemplo é um problema de confiabilidade relacionado a uma placa retangular feita de compósitos laminados, a ser resolvido utilizando o método de Monte Carlo, com 10^5 pontos amostrais. No problema em questão, a placa encontra-se simplesmente apoiada e com cargas no plano por unidade de comprimento N_x e N_{xy} como mostra a Figura 9. Será usada para solucionar o problema a teoria clássica de laminados como em BOGGELEN; LIMA(15, 16). Esta teoria define que: 1- linhas retas perpendiculares ao plano continuam retas e perpendiculares ao plano no estado deformado; 2- os segmentos normais à superfície são inextensíveis, ou seja, possuem espessura constante; 3- as forças transversais rotacionam com o plano de forma que continuem perpendiculares após sua deformação; 4- as forças são aplicadas na superfície média da placa. Isto pode ser observado na Figura 10.

Figura 9 – Visualização do exemplo 3.

A teoria de placas de Kirchhoff é a mais simples das teorias de equivalência de camada única para placas e é baseada no seguinte campo de deformações:

$$\begin{cases} u(x, y, z, t) = u_o(x, y, t) + z \frac{\delta w_o}{\delta x} \\ v(x, y, z, t) = v_o(x, y, t) + z \frac{\delta w_o}{\delta y} \\ w(x, y, z, t) = w_o(x, y, t) \end{cases} \quad (4.4)$$

Figura 10 – Teoria clássica de laminados.



Onde os deslocamentos ao longo das linhas de coordenadas de um material sobre o plano xy são definidos por u_o , v_o e w_o . Note que a forma do campo de deslocamentos permite a redução do problema com três dimensões para um estudo de deformação no plano de referência, como por exemplo no plano médio ($z = 0$). Uma vez que os deslocamentos no plano intermediário são conhecidos, os deslocamentos de qualquer ponto arbitrário no contínuo com três dimensões pode ser determinado usando a Equação 4.4. Assim, o problema é reduzido de três para duas dimensões.

O compósito em questão é composto de duas camadas de carbono epóxi, cada camada com 1,00 mm de espessura. As propriedades elásticas do material são $E_1 = 116600 \text{ MPa}$, $E_2 = 7673 \text{ MPa}$, $G_{12} = 4173 \text{ MPa}$ e $\nu_{12} = 0,27$. A orientação das camadas são $[0/45]$ como mostradas na Figura 9. Os esforços e propriedades do compósito serão considerados variáveis aleatórias com distribuição normal de acordo com a Tabela 3 e foram usados os mesmos valores de LOPEZ *et al.*(17). Foi considerado para este problema o critério de falha de Tsai-Wu. A Tabela 4 mostra os resultados do Monte Carlo para as sementes utilizadas neste estudo.

4.4.4 EXEMPLO 4

Neste exemplo será usada uma placa simplesmente apoiada como no exemplo 3, porém com esforços no plano de forças e momentos N_x , N_y e M_x como pode ser observado na Figura 11. Neste caso o laminado será composto de quatro camadas de carbono epóxi

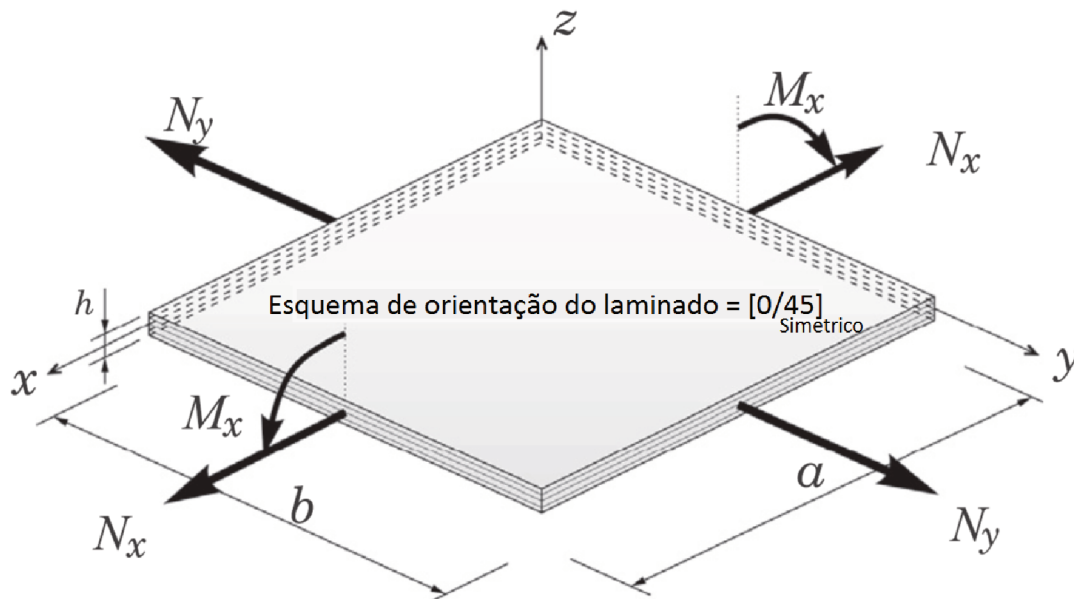
Tabela 3 – Valores para variáveis aleatórias do exemplo 3.

Número	Símbolo	Unidade	Média	COV
1	N_x	kN/m	300	0,20
2	N_{xy}	kN/m	100	0,20
3	X_t	MPa	2062	0,20
4	X_c	MPa	1701	0,20
5	Y_t	MPa	70	0,20
6	Y_c	MPa	240	0,20
7	S_{21}	MPa	105	0,20

Tabela 4 – Valores para o método de Monte Carlo do exemplo 3.

Identificador	Beta1	Beta2
<i>Seed</i> 0	3,08	1,11
<i>Seed</i> 1	3,14	1,10
Média(<i>Seed</i> 0 a 29)	3,10	1,11
Maior Beta(0 a 29)	3,16	1,12
Menor Beta(0 a 29)	3,03	1,10

de espessura 0,20 mm e terá as mesmas propriedades do material do exemplo anterior. Quanto a orientação das camadas, será utilizado [0/45/45/0]. Além disso, as orientações das camadas também serão consideradas como variáveis aleatórias de acordo com a Tabela 5.

Figura 11 – Visualização do exemplo 4.

Este problema é muito mais desafiador que o anterior, pois possui variáveis aleatórias com distribuição de Gumbel e lognormal, um maior número de variáveis aleatórias e, principalmente, porque a consideração de variabilidade na orientação das camadas impede

as simplificações em função da simetria não ser garantida. Para solucionar este problema foi utilizado o critério de falha da tensão limite máxima como estado limite.

Tabela 5 – Valores para variáveis aleatórias do exemplo 4.

Número	Símbolo	Unidade	Média	COV	Distribuição
1	N_x	kN/m	300,0	0,20	Gumbel
2	N_{xy}	kN/m	40,0	0,20	Gumbel
3	M_x	kNm/m	100,0	0,20	Gumbel
4	X_t	MPa	1500	0,20	Lognormal
5	X_c	MPa	1250	0,20	Lognormal
6	Y_t	MPa	50	0,20	Lognormal
7	Y_c	MPa	200	0,20	Lognormal
8	S_{21}	MPa	100	0,20	Lognormal
9	θ_1	°	0	0,20	Uniforme
10	θ_2	°	45	0,20	Uniforme
11	θ_3	°	45	0,20	Uniforme
12	θ_4	°	0	0,20	Uniforme

Tabela 6 – Valores para o método de Monte Carlo do exemplo 4.

Identificador	Beta1	Beta2	Beta3	Beta4
<i>Seed</i> 0	1,62	1,80	0,52	0,87
<i>Seed</i> 1	1,60	1,81	0,51	0,87
Média(<i>Seed</i> 0 a 19)	1,62	1,82	0,52	0,87
Maior Beta(0 a 19)	1,64	1,83	0,53	0,88
Menor Beta(0 a 19)	1,61	1,80	0,51	0,87

5 RESULTADOS

Inicialmente será analisado o funcionamento do programa com auxílio de gráficos, com isso será compreendido como o modelo de krigagem evolui em cada ciclo, depois serão avaliados os resultados de cada problema individualmente para uma comparação mais direta e por fim serão dadas as conclusões gerais dos resultados.

5.1 ANÁLISE GERAL DO PROGRAMA

Apesar das tabelas de resultados que serão mostradas serem suficientes para demonstrar o desempenho do programa, não é suficiente para compreendê-lo, por este motivo serão utilizados recursos gráficos. Para esta análise será observado o comportamento em cada ciclo do programa, como seria difícil de visualizar em mais de duas dimensões, serão utilizados apenas imagens para o exemplo 1.

Nas figuras coloridas dos resultados do modelo vale a seguinte legenda, que pode ser observada também nas legendas contidas em cada gráfico:

Linha vermelha tracejada - representa o valor limite da função (quando $G(v) = 0$), define o estado limite;

Pontos coloridos - cada um dos pontos amostrais, sendo usado um esquema de cores para representar o valor obtido pela aproximação do modelo de krigagem;

Pontos em formato quadrado - são os pontos da amostra com os melhores valores de aprendizado, que serão adicionados para o controle no próximo ciclo;

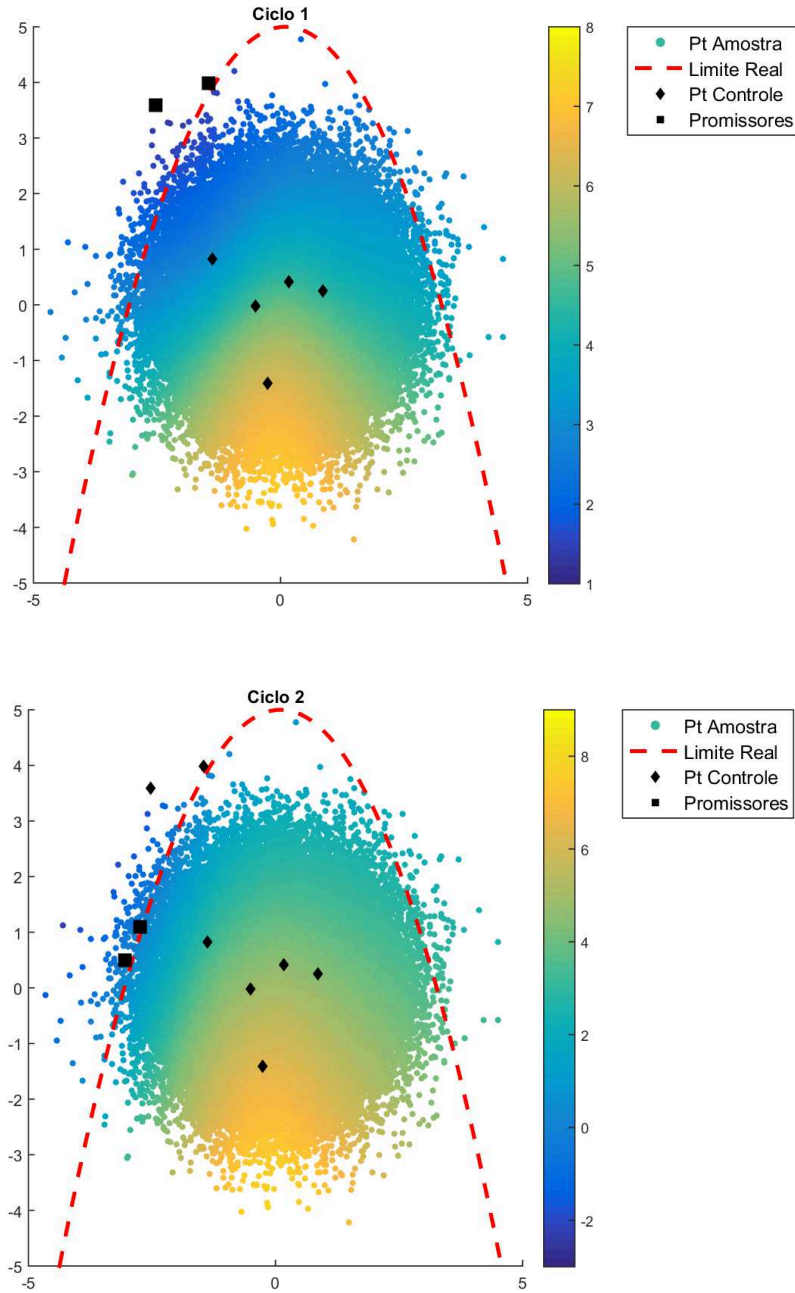
Pontos em forma de losango - pontos de controle usados para gerar o modelo de krigagem.

5.1.1 CASO 1 - NOÇÃO GERAL

O primeiro caso a ser analisado utiliza o modelo de krigagem com 5 pontos de controle iniciais, valor limite de aprendizagem igual a 1,5, semente de geração aleatória igual a 4 e adicionando 2 pontos de controle novos a cada ciclo para o exemplo 1. Nele será dada uma visão geral do funcionamento do programa.

Na figura 12, pode-se observar a alteração no resultado do modelo de krigagem conforme novos pontos são adicionados. No ciclo 1, a probabilidade de falha havia sido estimada como sendo zero, pois o menor valor estimado pela krigagem foi maior que 1. O programa observa então os melhores candidatos a pontos de controle por causa da alteração na condição de parada (em relação ao AK-MCS), a diferença de resultados é clara tanto no lado direito quanto no esquerdo para o segundo ciclo, sendo o valor mínimo encontrado reduzido para -2.

Figura 12 – Resultado do modelo para o caso 1. Parte 1



Na figura 13 pode-se observar quando o modelo começa a melhorar o lado esquerdo. No ciclo 9 um dos candidatos não está estimado tão próximo da fronteira de falha, porém considerando o erro de krigagem σ_{KO} ele tornou-se um bom candidato, portanto a inclusão deste termo na função de aprendizado foi de grande importância para o resultado final. Utilizando a figura 14 que mostra o gráfico de probabilidade de falha e coeficiente de confiabilidade, pode-se ver o efeito na probabilidade de falha, a estimativa sofre uma alteração drástica com apenas um ponto de controle novo, no primeiro momento o efeito pode ser negativo, porém com as adições de pontos no lado direito o modelo do ciclo 10 e

posteriores foram capazes de estimar bem toda a linha de falha.

Figura 13 – Resultado do modelo para o caso 1. Parte 2

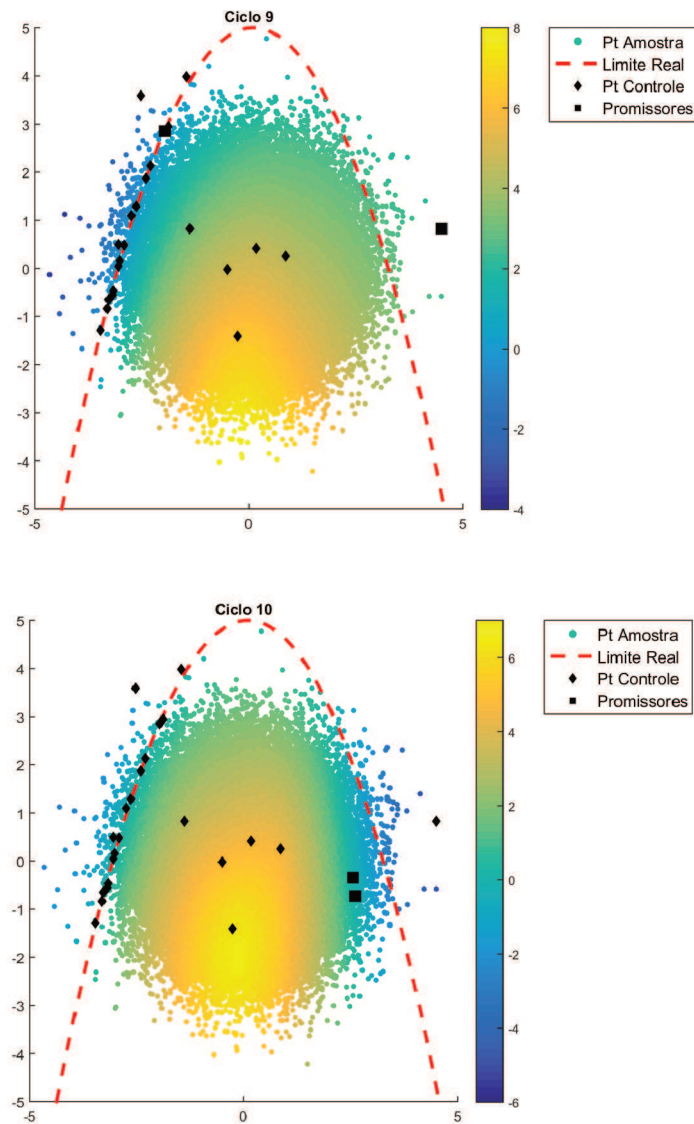


Figura 14 – Probabilidade de falha estimada para o caso 1.

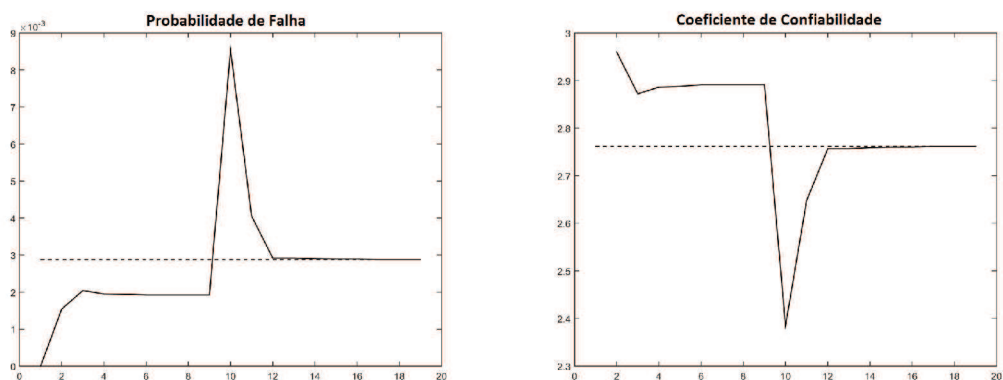
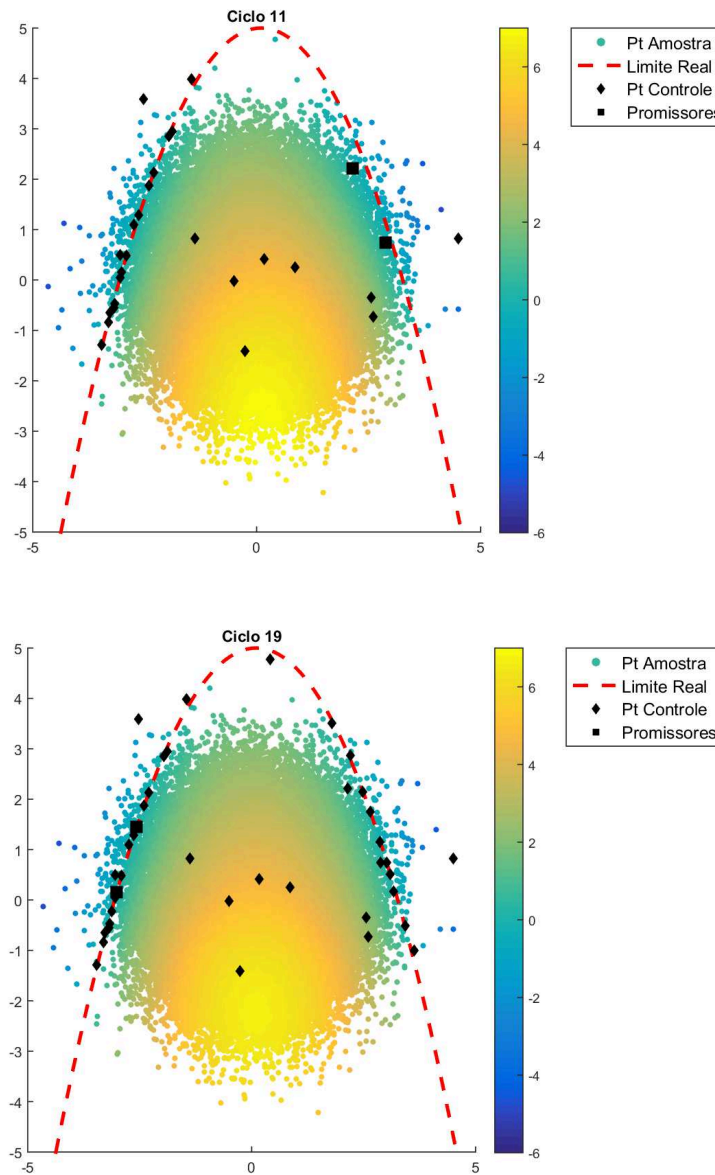


Figura 15 – Resultado do modelo para o caso 1. Parte 3



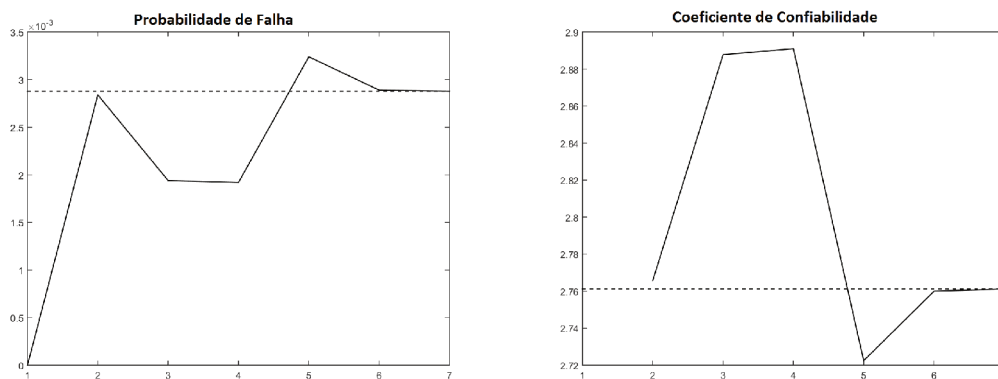
Nos gráficos da figura 14 a linha contínua representa os valores estimados, o eixo horizontal representa o número do ciclo, o eixo vertical a probabilidade de falha estimada e a linha tracejada representa a probabilidade de falha para a solução de Monte Carlo sem o uso da krigagem. O ponto de pico ocorre exatamente no ciclo 10, onde é estimado que a linha limite de falha esteja onde os dois pontos de melhor valor de aprendizagem que se encontram na segunda imagem da figura 13. Para visualizar como a probabilidade de falha volta a ser estimada de maneira correta, basta observar o ciclo 11 e 19 na figura 15. O ciclo 19 é o último ciclo, nele é possível ver que na situação final do modelo diversos pontos de controle estão posicionados na linha do limite de falha, o que é ideal para uma boa representação da falha.

5.1.2 CASO 2 - EFEITO DO NÚMERO DE PONTOS POR CICLO

No segundo caso, será estudado o efeito da alteração do número de pontos adicionados por ciclo, para isso será alterado o valor de pontos adicionados por ciclo do caso 1 de 2 para 10, possibilitando assim uma visualização por comparação.

Observa-se na figura 16 que o pico percebido no caso 1 se estende para dois ciclos, porém também é notável que ele ocorre pela primeira vez no ciclo 3 e está direcionado para a posição oposta.

Figura 16 – Probabilidade de falha estimada para o caso 2.



Nas imagens da figura 17 observa-se como o modelo inicia as estimações para o caso 2. A visualização do primeiro ciclo é praticamente igual ao caso 1, porém é visível que a partir do segundo ciclo o estimador avalia os resultados do lado direito com um erro crescente, no ciclo 4 inicia-se a melhoria do estimador na região de falha com poucos pontos até então, finalizando o processo de aprendizagem e resultando numa probabilidade com baixo erro no ciclo 7. Tanto o ciclo 4 quanto o 7 podem ser observados na figura 18.

Figura 17 – Resultado do modelo para o caso 2. Parte 1

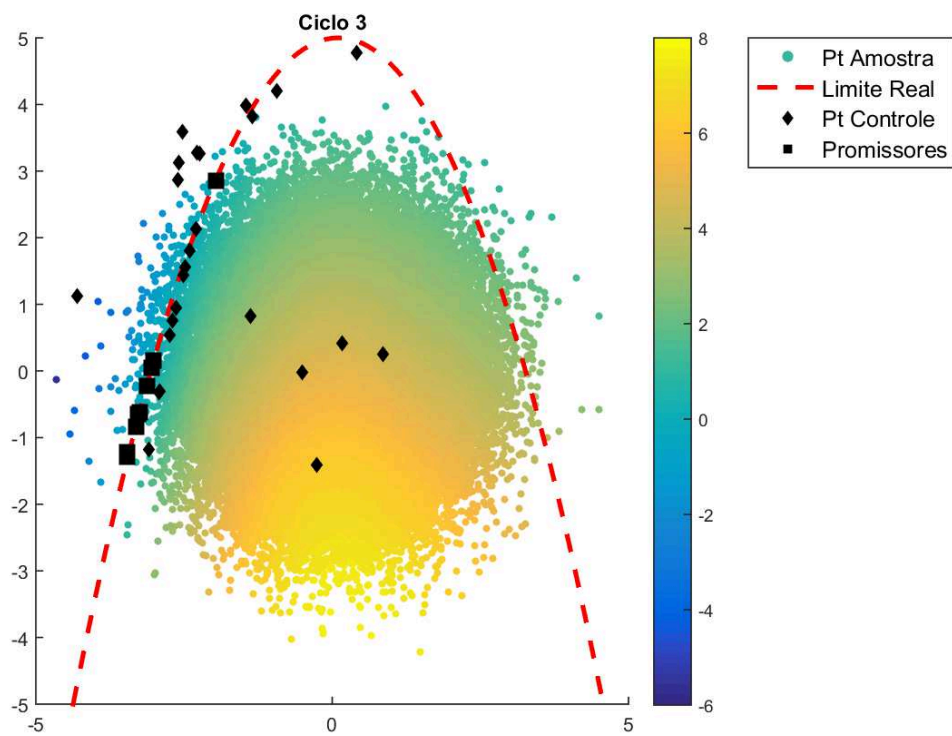
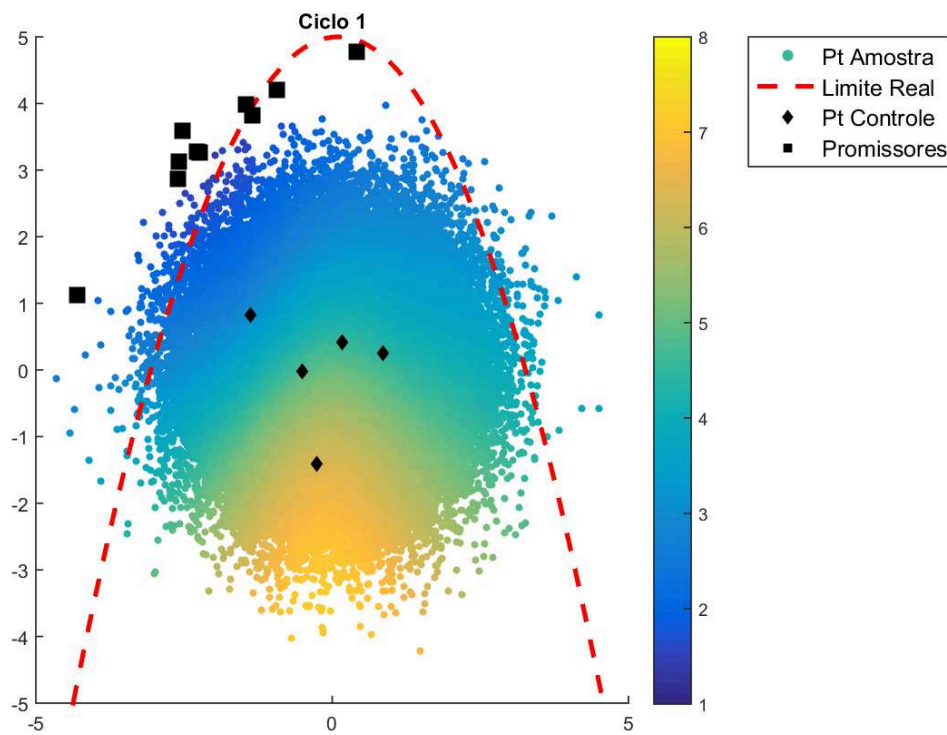
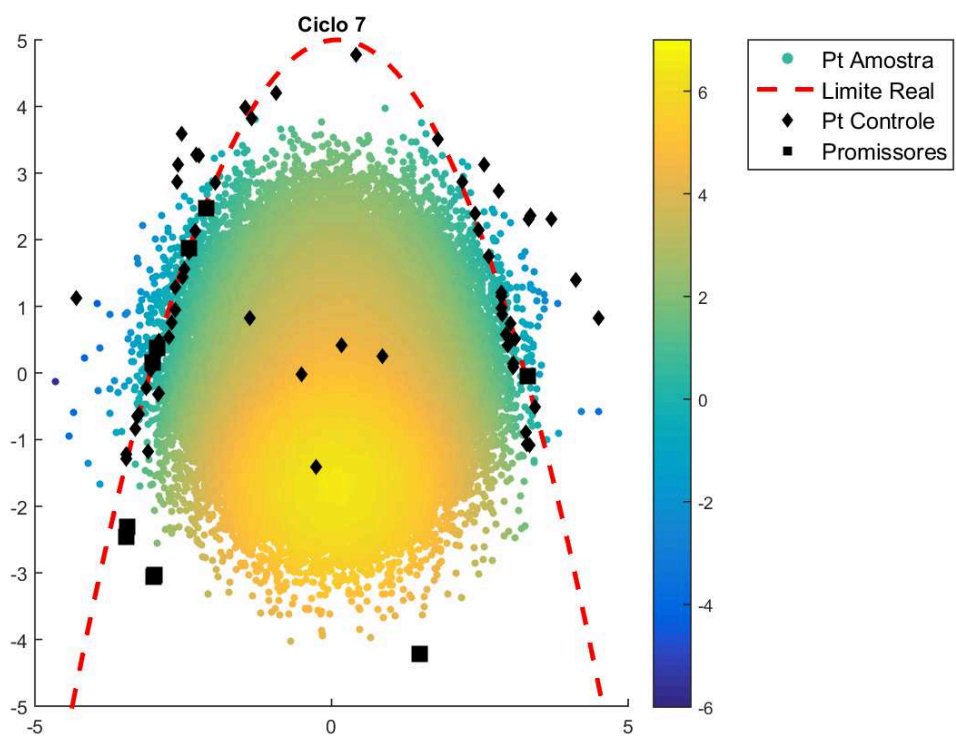
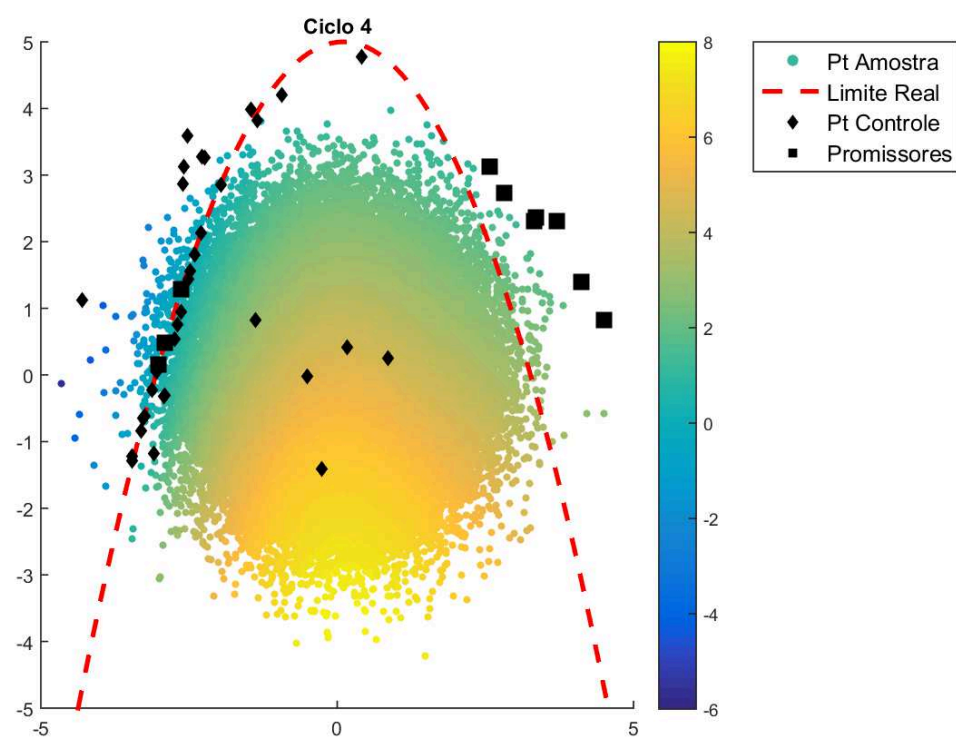


Figura 18 – Resultado do modelo para o caso 2. Parte 2

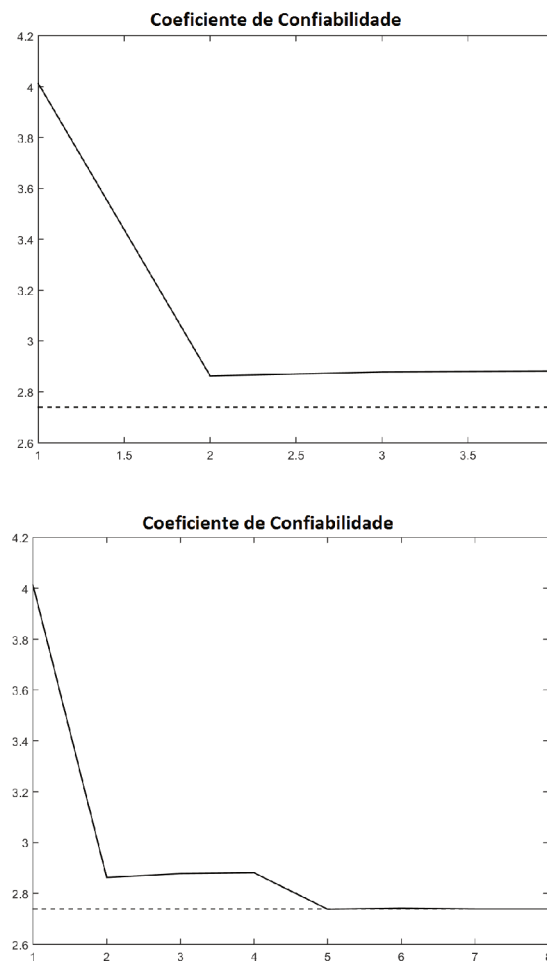


5.1.3 CASO 3 - EFEITO DO VALOR DE APRENDIZAGEM

Para estudar este caso, serão apresentadas duas situações, ambas utilizando 10 pontos de controle iniciais, adicionando 10 pontos novos por ciclo do programa, com a mesma semente de geração aleatória, sendo ela de valor 1, porém os valores limites de aprendizagem serão diferentes. Para a primeira situação será utilizado o valor de aprendizagem $U_{lim} = 1,5$, entretanto, para a segunda situação será utilizado o valor limite de aprendizagem igual a 3.

Primeiramente pode-se observar nos gráficos da figura 19 a comparação do resultado para o coeficiente de confiabilidade β encontrado para a primeira e para a segunda situação. Nota-se que no primeiro gráfico só foi computado até o quarto ciclo, entretanto, no segundo gráfico o quinto ciclo possui um valor mais adequado para o coeficiente de confiabilidade, além disso o programa procede até o oitavo ciclo.

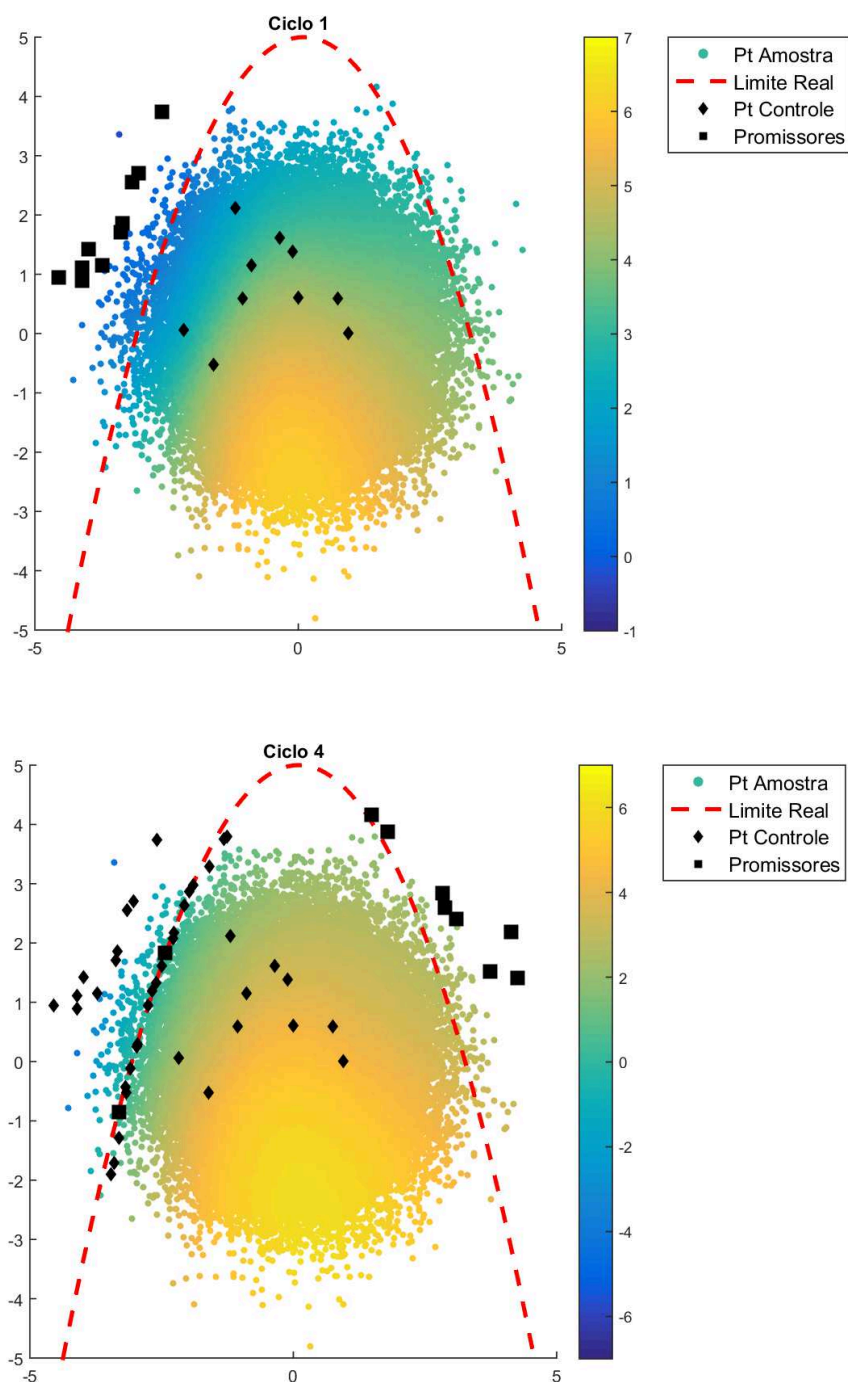
Figura 19 – Probabilidade de falha estimada para o caso 3.



Para compreender a progressão destas duas situações, deve-se observar sua sequência e o motivo de parada da primeira e segunda situação serem em ciclos tão distantes, assim como o motivo do resultado final ter tamanha diferença. Na figura 20 pode-se observar

tanto a primeira quanto a segunda situação simultaneamente, sabendo que ambas tem as mesmas condições iniciais impostas pelo programa, ambas progridem exatamente da mesma forma.

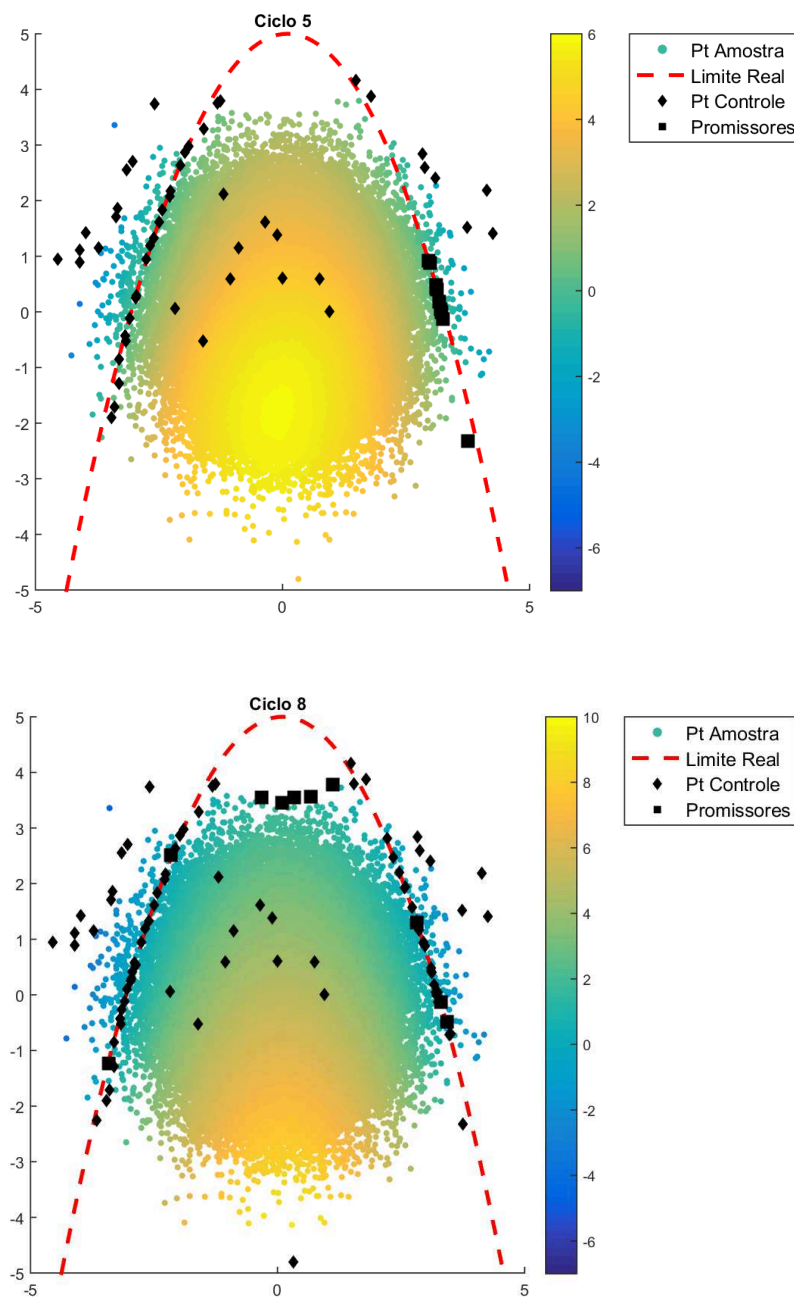
Figura 20 – Resultado do modelo para o caso 3. Parte 1



Após o ciclo 4, no qual se observa na figura 20 que a região de falha à direita começa a ser melhor estimada, as duas situações passam a ser diferentes. É fácil perceber que o motivo é o critério de parada de aprendizagem, para confirmar isto, foi avaliado o melhor

valor de aprendizagem para o quarto ciclo, sendo este valor igual a 1,9 aproximadamente. Considerando que apenas esse critério de parada foi utilizado, para a primeira situação com $U_{lim} = 1,5$ considerou-se que o modelo estava suficientemente bom para mostrar o resultado final, chegando a um erro em relação ao valor sem o uso da krigagem muito maior que para a segunda situação. Para avaliar rapidamente a evolução do modelo para a segunda situação, na figura 21 pode-se observar os ciclos 5 e 8.

Figura 21 – Resultado do modelo para o caso 3. Parte 2



5.2 ANÁLISE DE CADA EXEMPLO

5.2.1 EXEMPLO 1

Primeiramente será feita uma avaliação do exemplo 1, que é o mais simples dos problemas e depende apenas de duas variáveis aleatórias. As afirmações feitas para este exemplo são válidas a princípio apenas para o mesmo, tais afirmações serão confirmadas ou refutadas por meio dos outros casos. Como foram testados aproximadamente 700 valores distintos para as configurações do programa desenvolvido serão mostrados apenas alguns dos resultados de interesse. Foram analisadas variações para o número de pontos iniciais, o número de pontos adicionados por ciclo, a semente de geração dos números aleatório e o valor limite de aprendizado.

É possível identificar na Tabela 7 que o número inicial de pontos pode ter pouca ou nenhuma influência no valor final de confiabilidade, mas influencia o número de pontos no final dos ciclos. Nota-se que os valores da probabilidade de falha ficaram praticamente inalterados em todos os valores mostrados. Para esta tabela é possível perceber também que o número total de pontos adicionados ficou entre 50 a 80 aproximadamente com poucas exceções. É notável a consistência do resultado para este caso estudado.

Tabela 7 – Dados do programa com krigagem. Exemplo 1, semente 0, Ulim 1,5.

Ninit	Nfinal	Ulim	$P_f(\times 10^{-3})$	Erro P_f	Beta	ErroBeta	Seed	nAdd
5	64	1,5	2,925	0,0362	2,756	0,0044	0	1
50	112	1,5	2,925	0,0362	2,756	0,0044	0	1
200	250	1,5	2,925	0,0362	2,756	0,0044	0	1
300	350	1,5	2,925	0,0362	2,756	0,0044	0	1
5	75	1,5	2,925	0,0362	2,756	0,0044	0	5
50	120	1,5	2,925	0,0362	2,756	0,0044	0	5
200	255	1,5	2,925	0,0362	2,756	0,0044	0	5
300	355	1,5	2,925	0,0362	2,756	0,0044	0	5
5	95	1,5	2,925	0,0362	2,756	0,0044	0	10
50	130	1,5	2,925	0,0362	2,756	0,0044	0	10
200	260	1,5	2,925	0,0362	2,756	0,0044	0	10
300	390	1,5	2,925	0,0362	2,756	0,0044	0	10

O valor final varia principalmente em função da qualidade dos pontos e não apenas da quantidade, é possível ter "azar" nos pontos iniciais tornando a estimativa dos melhores valores de aprendizado ruins, pode-se observar na tabela 8 que um menor número de pontos iniciais deixa o estimador mais dependente do gerador de aleatórios, pois os pontos podem ser sorteados todos numa mesma região.

É possível observar também que o número de pontos adicionados em cada iteração do programa afeta o número de pontos no final da execução assim como altera um pouco o valor final da confiabilidade encontrada. Isso se dá pelo fato de que pegar os $nAdd$

melhores pontos, inclui pontos que poderiam não ser necessários. Por exemplo: se existem 3 pontos com valores de aprendizado bons e muitos pontos ruins, com $nAdd = 1$ apenas o melhor seria escolhido, podendo melhorar ou piorar o valor de aprendizado dos outros pontos, porém com $nAdd = 10$ os 10 melhores serão adicionados como pontos de controle para a krigagem, independente de cada um respeitar o valor de aprendizado limite desde que o melhor respeite. Apesar de isso não garantir que a quantidade de pontos adicionados por ciclo influencie os resultados, considerando que é possível pontos ruins piorarem a estimativa da região de falha, na prática afeta diretamente o número de pontos de controle aumentando a chance de um bom resultado. Por este motivo um maior número de pontos adicionados por ciclo é vantajoso de forma geral, até um limite.

Tabela 8 – Dados do programa com krigagem. Exemplo 1, semente 3, Ulim 1,5.

Ninit	Nfinal	Ulim	$P_f(\times 10^{-3})$	Erro P_f	Beta	ErroBeta	Seed	nAdd
5	34	1,5	2,042	0,3207	2,872	0,0453	3	1
8	36	1,5	2,042	0,3207	2,872	0,0453	3	1
10	38	1,5	2,042	0,3207	2,872	0,0453	3	1
50	107	1,5	3,096	0,0299	2,737	0,0035	3	1
200	249	1,5	3,097	0,0303	2,737	0,0036	3	1
5	37	1,5	2,042	0,3207	2,872	0,0453	3	2
8	42	1,5	2,042	0,3207	2,872	0,0453	3	2
10	38	1,5	2,042	0,3207	2,872	0,0453	3	2
50	114	1,5	3,096	0,0299	2,737	0,0035	3	2
200	250	1,5	3,096	0,0299	2,737	0,0035	3	2
5	45	1,5	2,042	0,3207	2,872	0,0453	3	5
8	48	1,5	2,042	0,3207	2,872	0,0453	3	5
10	45	1,5	2,042	0,3207	2,872	0,0453	3	5
50	115	1,5	3,096	0,0299	2,737	0,0035	3	5
200	255	1,5	3,096	0,0299	2,737	0,0035	3	5
5	105	1,5	3,096	0,0299	2,737	0,0035	3	10
8	108	1,5	3,096	0,0299	2,737	0,0035	3	10
10	110	1,5	3,096	0,0299	2,737	0,0035	3	10
50	140	1,5	3,097	0,0303	2,737	0,0036	3	10
200	270	1,5	3,096	0,0299	2,737	0,0035	3	10

Ressalta-se que um número maior de pontos adicionados por ciclo não significa um maior número de ciclos do programa, uma vez que sair de 5 pontos e terminar com 64 adicionando de 1 em 1 requer 59 ciclos, mas de 5 para 95 pontos são apenas 9 ciclos com $nAdd = 10$. Nota-se inclusive que nas tabelas apresentadas o contrário é verdade, quanto maior o $nAdd$, menor o número de ciclos necessários, mesmo com um número final de pontos de controle maior. Lembrando da figura 7 anterior, esta é a vantagem do método proposto em relação ao AK-MCS.

No geral a tabela 9 mostra que o U_{lim} não tem quase nenhum feito além do número final de pontos para o exemplo 1. Logicamente, caso o valor limite de aprendizado seja excessivamente baixo, um menor número de pontos será usado, podendo ser insuficiente para uma boa representação da região de falha.

Recomenda-se que o valor de U_{lim} seja por volta de 2,5, mas caso seja necessária a garantia do resultado deve-se usar valores maiores. Lembrando que essa condição foi observada para o exemplo 1, além disso, os resultados mostrados foram aqueles em que o U_{lim} tiveram maior efeito sobre o erro encontrado. No total foram 32 casos distintos testados para valores limite de aprendizado extremos e em apenas três o $U_{lim} = 0,5$ teve pior desempenho em relação ao $U_{lim} = 2,5$. Como ocorreram casos de pequena variação do erro melhores para o $U_{lim} = 0,5$ e casos melhores para o $U_{lim} = 2,5$, essas pequenas variações não foram contadas, afinal existe aleatoriedade envolvida.

Tabela 9 – Dados do programa com krigagem. Exemplo 1, Variando Ulim

Ninit	Nfinal	Ulim	$P_f(\times 10^{-3})$	Erro P_f	Beta	ErroBeta	Seed	nAdd
5	33	0,5	1,874	0,3825	2,899	0,0564	0	1
5	37	1	1,874	0,3825	2,899	0,0564	0	1
5	64	1,5	2,925	0,0362	2,756	0,0044	0	1
5	72	2,5	2,925	0,0362	2,756	0,0044	0	1
5	79	3,5	2,925	0,0362	2,756	0,0044	0	1
5	88	5	2,925	0,0362	2,756	0,0044	0	1
5	109	10	2,925	0,0362	2,756	0,0044	0	1
5	29	0,5	2,042	0,3207	2,872	0,0453	3	1
5	33	1	2,042	0,3207	2,872	0,0453	3	1
5	34	1,5	2,042	0,3207	2,872	0,0453	3	1
5	40	2,5	2,042	0,3207	2,872	0,0453	3	1
5	40	3,5	2,042	0,3207	2,872	0,0453	3	1
5	74	5	3,096	0,0299	2,737	0,0035	3	1
5	92	10	3,096	0,0299	2,737	0,0035	3	1
50	130	0,5	3,097	0,0303	2,737	0,0036	3	10
50	130	1	3,097	0,0303	2,737	0,0036	3	10
50	140	1,5	3,097	0,0303	2,737	0,0036	3	10
50	140	2,5	3,097	0,0303	2,737	0,0036	3	10
50	150	3,5	3,096	0,0299	2,737	0,0035	3	10
50	160	5	3,096	0,0299	2,737	0,0035	3	10
50	170	10	3,096	0,0299	2,737	0,0035	3	10

5.2.2 EXEMPLO 2

Para o exemplo 2, foram testados 72 casos com as seguintes variações:
 $U_{lim} = 1,5; 2,5; 5;$

$N_{init} = 5; 50; 100; 200;$

$n_{Add} = 1; 5; 10;$

$Seed = 0; 1.$

Infelizmente, não ocorreu nenhuma variação notável nos erros em relação a uma mesma semente de geração dos números aleatórios. Logicamente, não haver alteração não significa incapacidade de analisar os dados, pelo contrário, demonstra que possivelmente as seguintes afirmações do exemplo 1 são válidas:

- 1 - O U_{lim} não deve ter grande impacto na probabilidade de falha estimada;
- 2 - O n_{Add} pode auxiliar na garantia de um bom resultado, mas de maneira geral apenas altera o número de pontos de controle no final do programa;
- 3 - O N_{init} tem pouca a nenhuma influência no valor final, dependendo principalmente da qualidade dos pontos sorteados, considerando que não teve efeito para este exemplo.

Entretanto a tabela 10 é mostrada para observar a variação do número final de pontos e demonstrar a afirmação de baixa alteração no resultado final.

Tabela 10 – Dados do programa com krigagem. Exemplo 2.

Ninit	Nfinal	Ulim	$P_f(\times 10^{-2})$	Erro P_f	Beta	ErroBeta	Seed	nAdd
5	149	1,5	1,177	0,0222	2,264	0,0038	0	1
5	179	2,5	1,177	0,0222	2,264	0,0038	0	1
5	235	5	1,177	0,0222	2,264	0,0038	0	1
5	165	1,5	1,177	0,0222	2,264	0,0038	0	10
5	185	2,5	1,177	0,0222	2,264	0,0038	0	10
5	245	5	1,177	0,0222	2,264	0,0038	0	10
5	198	1,5	1,198	0,0117	2,258	0,0020	1	1
50	193	1,5	1,198	0,0117	2,258	0,0020	1	1
100	224	1,5	1,198	0,0117	2,258	0,0020	1	1
200	318	1,5	1,198	0,0117	2,258	0,0020	1	1
5	215	1,5	1,198	0,0117	2,258	0,0020	1	10
50	200	1,5	1,198	0,0117	2,258	0,0020	1	10
100	240	1,5	1,198	0,0118	2,258	0,0020	1	10
200	340	1,5	1,198	0,0117	2,258	0,0020	1	10

5.2.3 EXEMPLO 3

O exemplo 3 é o primeiro problema de confiabilidade baseado diretamente em um problema real. Portanto, as análises deste exemplo possuem maior semelhança com possíveis usos práticos.

Pode-se observar com a tabela 11 que o número de pontos de controle iniciais para o modelo de krigagem afetaram constantemente o resultado, sendo no geral, um

maior número de pontos melhor. Infelizmente por problemas de custo computacional, o computador utilizado para estes testes não conseguia processar o sistema de equações para mais de 200 pontos de controle, mas ainda é possível afirmar que a estimativa inicial dos resultados é crucial para descrever bem a probabilidade de falha. Baseado nos estudos realizados, o maior motivo desta necessidade de pontos iniciais para este problema vem da quantidade de variáveis aleatórias envolvidas, que neste caso são 7.

Tabela 11 – Dados do programa com krigagem. Exemplo 3, *Seed* 0.

Ninit	Nfinal	Ulim	Beta1	ErroBeta1	Beta2	ErroBeta2	Seed	nAdd
5	127	1,5	3,079	0,0009	1,251	0,1251	0	1
10	146	1,5	3,076	0,0019	1,102	0,0090	0	1
25	151	1,5	3,076	0,0019	1,087	0,0225	0	1
50	183	1,5	3,079	0,0009	1,121	0,0089	0	1
100	200	1,5	3,079	0,0009	1,120	0,0074	0	1
150	200	1,5	3,070	0,0037	1,098	0,0118	0	1
5	175	1,5	3,076	0,0019	1,205	0,0837	0	10
10	180	1,5	3,079	0,0009	1,187	0,0675	0	10
25	155	1,5	3,076	0,0019	1,146	0,0312	0	10
50	150	1,5	3,079	0,0009	1,096	0,0143	0	10
100	200	1,5	3,079	0,0009	1,107	0,0045	0	10
150	200	1,5	3,041	0,0132	1,089	0,0204	0	10
5	180	2,5	3,079	0,0009	1,292	0,1619	0	5
10	150	2,5	3,076	0,0019	1,163	0,0465	0	5
25	150	2,5	3,079	0,0009	1,142	0,0270	0	5
50	150	2,5	3,079	0,0009	1,107	0,0038	0	5
100	200	2,5	3,079	0,0009	1,113	0,0015	0	5
150	200	2,5	3,073	0,0028	1,116	0,0038	0	5

Tabela 12 – Comparação para o exemplo 3.

Método	Beta1	erroBeta1	Beta2	erroBeta2	pontosAvaliados
MelhorB1	3,14	0,0000	1,13	0,0270	200
MelhorB2	3,08	0,0019	1,11	0,0000	202
PiorB1	3,04	0,0132	1,09	0,0204	200
PiorB2	3,08	0,009	1,44	0,2947	161
Média	3,09	0,0050	1,15	0,0444	171
LOPEZ <i>et al.</i> (17) Uniforme	inf	inf	1,020	0,0060	25
LOPEZ <i>et al.</i> (17) Lognormal	3,050	0,0066	1,106	0,0006	25
FORM Uniforme	inf	inf	0,89	0,1190	157
FORM Lognormal	3,11	0,0144	0,89	0,0353	171
Monte Carlo	3,03	3,16	1,10	1,12	100000

Este problema será comparado por meio dos dados de LOPEZ *et al.*(17), que incluem os valores do método do FORM (*First-Order Reliability Method* explicado em muitas literaturas, como em BECK(1)). Também foram colocados os menores e maiores valores encontrados para Beta1 e Beta2 usando a simulação de Monte Carlo simples, que são os valores que aparecem na linha inferior da tabela 12. Nesta tabela pode-se notar que o pior valor para o Beta1 foi muito melhor que o encontrado pelos outros métodos, chegando a 0% em relação à simulação de Monte Carlo. Além disso, foi possível encontrar o valor de Beta1 em todos os casos testados, uma vantagem em relação aos outros métodos. No caso do Beta2, o pior resultado encontrado nesse estudo chegou a quase 30% de erro em relação com o Monte Carlo utilizando dos mesmos pontos amostrais, sendo 30% de erro inadequado para confiabilidade.

No estudo usado para comparação os valores foram encontrados individualmente, porém neste trabalho, mesmo utilizando um modelo de krigagem para cada, foram usados os mesmo pontos de controle para Beta1 e Beta2 com o objetivo de reduzir o tempo de processamento. Juntando essa limitação de usar os mesmos pontos de controle com a limitação computacional de 200 pontos é possível afirmar que resultados melhores poderiam ser obtidos.

Apesar de tudo o melhor resultado conseguiu chegar a erros inferiores a 0,01%, tornando o método estudado neste trabalho o melhor resultado comparado. Enquanto avaliando os valores médios, para o coeficiente de confiabilidade 1 chegou-se a resultados bons em relação ao FORM, ao método proposto por LOPEZ *et al.*(17) e ao Monte Carlo simples, entretanto para o segundo coeficiente o programa desenvolvido foi insatisfatório.

5.2.4 EXEMPLO 4

Este exemplo, além de ser mais complexo, possui 4 equações de estado limite resolvidas de maneira independente, sendo um problema usar os mesmos pontos de controle para todos os modelos de krigagem e ainda considerando a capacidade computacional disponível para resolver os problemas, limitando os pontos de controle a 200. Todos os casos testados chegaram a esse limite de 200, logo o valor de aprendizado real é inferior a 1,5, que foi o menor testado, mostrando que nenhum resultado testado utilizou todo o potencial do método desenvolvido.

Para este caso, será apenas apresentada uma breve tabela comparativa com os dados encontrados por LOPEZ *et al.*(17), apesar de outros autores terem resolvido o mesmo problema, não foi comparado em função do limitante comentado.

Os resultados no geral foram piores que os valores de comparação como era de se esperar. É certo afirmar que os resultados seriam melhores se fossem divididos em quatro problemas separados (sem usar os mesmos pontos de controle) e não existisse a limitação computacional de 200 pontos. Entretanto, é difícil afirmar que seriam bons resultados em

função da dimensão do vetor de variáveis de entrada, neste caso composto por 13 variáveis. É possível que este seja um defeito relacionado ao programa desenvolvido.

Tabela 13 – Comparação para o exemplo 4.

Método	erroBeta1	erroBeta2	erroBeta3	erroBeta4
MelhorB1	0,0002	0,0056	0,2296	0,0149
MelhorB2	0,0261	0,0030	0,0980	0,0816
MelhorB3	0,0225	0,0030	0,0004	0,0816
MelhorB4	0,0095	0,0386	0,0727	0,0117
PiorB1	0,0498	0,0893	0,1771	0,0425
PiorB2	0,0174	0,0961	0,0838	0,0776
PiorB3	0,0156	0,0790	0,3492	0,0643
PiorB4	0,0039	0,0876	0,3233	0,4303
Média	0,0158	0,0340	0,1276	0,0918
LOPEZ <i>et al.</i> (17)	0,0032	0,0117	0,0198	0,0014

6 CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS

Neste trabalho, foi apresentado um novo procedimento para o uso do modelo de krigagem em problemas de confiabilidade, além de apresentar os conceitos necessários para compreender o modelo e o procedimento. Tal modelo foi analisado por meio de problemas simples de confiabilidade e feito o auxílio gráfico com o objetivo de entender o processo, explicar os resultados e planejar melhorias.

De maneira geral, o programa desenvolvido foi suficiente para resolver os problemas propostos. Para os exemplos 1 e 2, os erros no índice de confiabilidade computados ficaram por volta de 1% ou menos. Para o exemplo 3, onde atingiu-se o limite de pontos de controle algumas vezes, o erro variou de quase 0% a até 30%, entretanto o erro médio foi de 0,5% para o coeficiente de confiabilidade 1 e de 4,44% para o coeficiente 2, que é aceitável na maioria dos casos. Finalmente, para o exemplo 4, do qual todos os valores testado atingiram o limite de pontos de controle, o erro dos coeficientes de confiabilidade variou de 0,02% a 43%, sendo em média um erro de 1,6%, 3,4%, 13% e 9% para os índices de confiabilidade relacionados às equações de estado limite de 1 a 4 respectivamente, tais valores não são bons. Entretanto estima-se que o aumento na quantidade de pontos de controle (impedido pelo limite computacional) teria melhorado os resultados de maneira considerável. Além disso, até mesmo a simulação de Monte Carlo possuiu variações de acordo com a semente usada para o gerador de números aleatórios, portanto pode-se concluir que o programa está cumprindo com seu objetivo de estimar os valores de falha e confiabilidade de maneira aceitável, deve-se apenas encontrar um modo de reduzir o tamanho das equações para a otimização dos pesos do modelo de krigagem.

Infelizmente a aleatoriedade do programa incapacitou afirmar com certeza quais os melhores valores para cada variável configurável do programa, mas estima-se que um número não muito alto de pontos iniciais seja a melhor escolha (por volta de 10 a 50), um número médio de novos pontos por ciclo (recomendado por volta de 5) e um valor de aprendizado suficiente para estimar todo o limite de falha (recomenda-se por volta de 2,5 a 5).

Apesar deste método cumprir com seu objetivo, conclui-se que existe uma grande variedade de opções para melhorá-lo. Alguns exemplos são: melhor caracterização inicial por meio dos primeiros pontos de controle, utilizar um critério de aprendizado melhor ou até mesmo reduzir a quantidade de pontos que influenciam as regiões de estudo no modelo de krigagem. Além disso, seria possível testar os efeitos da modelagem por krigagem na simulação de Monte Carlo por importância que tende a usar apenas pontos próximos à região de falha.

Por fim, espera-se em trabalhos futuros:

1. Melhorar a distribuição dos pontos de controle iniciais. Atualmente, utilizar muitos pontos iniciais gera aglomerações, sendo que essa aglomeração acrescenta muito pouco para o modelo de krigagem e aumenta o custo computacional;
2. Estudar mais sobre a função de aprendizado. Com um melhor conhecimento sobre os possíveis jeitos de classificar os pontos amostrais seria possível não só melhorar a escolha do próximo ponto, como talvez deixar a própria função de aprendizado definir se há ou não necessidade de acrescentar mais pontos de controle, prevendo, por exemplo, regiões mal estimadas (problema ilustrado nas figuras 19, 20 e 21);
3. Melhorar a escolha dos novos pontos de controle. Além da função de aprendizado, seria importante impedir a aglomeração dos novos pontos de controle. Basta observar as figuras que demonstram os pontos promissores para 10 novos pontos por ciclo que é fácil ver as aglomerações. Como dito anteriormente, aglomerações tem pouco ganho para o modelo e ainda aumentam o custo computacional;
4. Alteração dos pontos usados no estimador. O modelo de krigagem ordinária, usado neste trabalho, usa todos os pontos conhecidos para gerar um sistema de equações. Considerando que alguns casos possuíam 200 pontos de controle, significa 201 variáveis por modelo de krigagem feito. Observando que o sistema de equações é feito para cada ponto da amostra, obtêm-se um sistema desnecessariamente extenso, repetido quase 1 milhão de vezes. A soma de todos os pesos deve resultar em 1, assim como apenas os grupos de pontos próximos ganham pesos consideráveis, logo, propõem-se desenvolver um método para definir quais pontos conhecidos serão usados para gerar as equações do modelo de krigagem.

Para finalizar, verificou-se que este método pode ser utilizado para problemas de confiabilidade. Porém, para evitar os problemas verificados neste estudo, seria necessário estudar e testar as alterações propostas. É quase certo que com um bom desenvolvimento destas, seria possível fazer um método rápido e confiável para agilizar e reduzir o custo computacional da simulação de Monte Carlo.

REFERÊNCIAS

- 1 BECK, A. T. *Curso de Confiabilidade Estrutural*. São Paulo: Escola de engenharia de São Carlos, 2014. Citado 4 vezes nas páginas 15, 18, 27 e 62.
- 2 CHOI, S.; GRANDHI, R.; CANFIELD, R. *Reliability-based Structural Design*. EUA: Springer Science & Business Media, 2007. Citado na página 17.
- 3 PAPOULIS, A.; PILLAI, S. U. *Probability, random variables, and stochastic processes*. 4. ed. EUA: Tata McGraw-Hill Education, 2002. Citado na página 17.
- 4 SUDRET, B. *Uncertainty propagation and sensitivity analysis in mechanical models Contributions to structural reliability and stochastic spectral methods*. França: Universidade Blaise Pascal, 2007. Citado na página 17.
- 5 MONTGOMERY, D. C.; RUNGER, G. C.; HUBELE, N. F. *Estatística aplicada e probabilidade para engenheiros*. 2. ed. Rio de Janeiro: LTC, 2003. Citado na página 17.
- 6 SILVA, P. A. L. D. *Probabilidades & estatística*. Rio de Janeiro: Reichmann & Affonso, 1999. Citado na página 17.
- 7 REDDY, T. A. *Applied data analysis and modeling for energy engineers and scientists*. EUA: Springer Science & Business Media, 2011. Citado na página 17.
- 8 ROSS, S. M. *Introduction to Probability Models*. 9. ed. EUA: Academic Press, 2007. Citado na página 20.
- 9 OLEA, R. A. *Geostatistics for engineers and earth scientists*. Kansas: Kluwer Academic Publishers, 1999. Citado 2 vezes nas páginas 29 e 32.
- 10 LUENBERGER, D. *Optimization by Vector Space Methods*. New York: John Wiley & Sons, 1969. Citado na página 29.
- 11 JOURNAL, A.; HUIJBREGTS, C. *Mining Geostatistics*. New Jersey: The Blackburn Press, 1978. Citado 2 vezes nas páginas 29 e 32.
- 12 KROETZ, H. M. *Meta-modelagem em confiabilidade estrutural*. Dissertação (Mestrado em Engenharia Civil) — Universidade de São Carlos, São Paulo, 2015. Citado na página 29.
- 13 CORREIA, P. *Modelação e Estimação, uma introdução à geoestatística*. Paraná: Laboratório de Estatística e Geoinformação, UFPR, 2010. Citado na página 29.
- 14 ECHARD, B.; GAYTON, N.; LEMAIRE, M. *AK-MCS: An Active Learning Reliability Method Combining Kriging and Monte Carlo Simulation*. França: Clermont Université, Institut Français de Mécanique Avancée, 2010. Citado na página 37.
- 15 BOGGELEN, D. V. *Failure Analysis of Fiber Reinforced Laminated Composite Materials, Using Analytical and FEM Techniques*. Bacharelado em Engenharia — Hanze University Groningen, Holanda, 2006. Citado na página 43.

- 16 LIMA, M. V. A. *Análise de Placas Laminadas pelo Método das Diferenças Finitas Energéticas*. Trabalho de Conclusão de Curso de Engenharia Civil — Universidade Federal de Feira de Santana, 2010. Citado na página 43.
- 17 LOPEZ, R. *et al.* Advantages of employing a full characterization method over form in the reliability analysis of laminated composite plates. 2014. Citado 5 vezes nas páginas 42, 44, 61, 62 e 64.